AD-A055 425 NAVAL POSTGRADUATE SCHOOL MONTEREY CALIF F/G 12/1

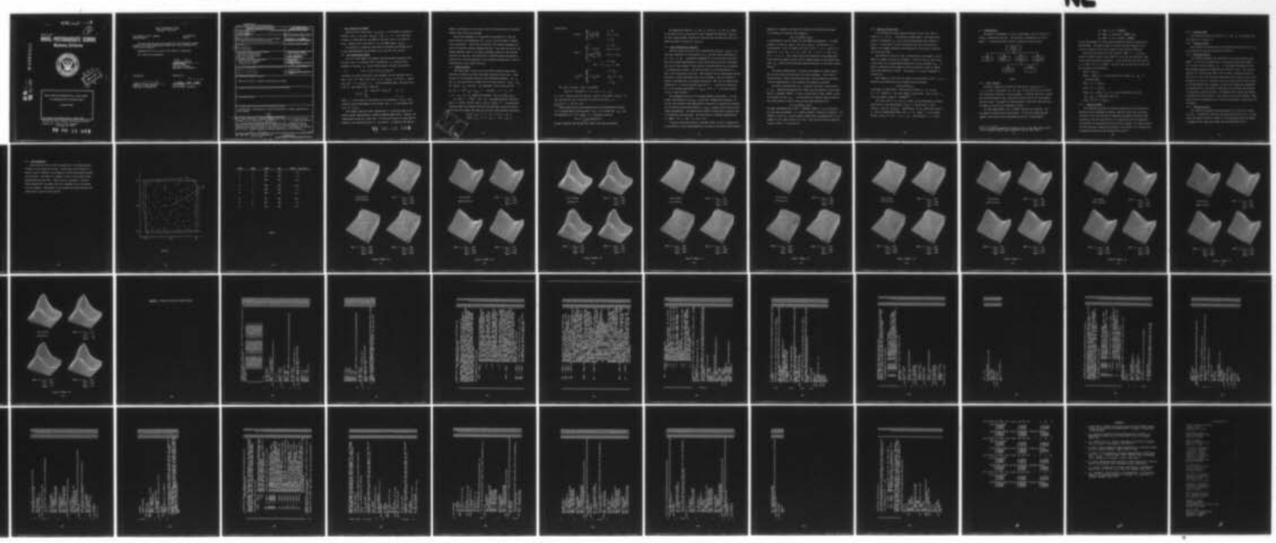
SMOOTH SURFACE APPROXIMATION BY A LOCAL METHOD OF INTERPOLATION--ETC(U)

APR 78 . R FRANKE

UNCLASSIFIED NPS53-78-002 NL

| OF |

AD
A055425

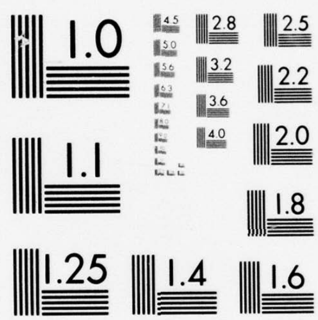


END

DATE FILMED

8 -78

DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

FOR FURTHER TRAN

NPS53-78-002

NAVAL POSTGRADUATE SCHOOL

Monterey, California

AD A055425

AD No.

DDC FILE COPY

DDC
JUN 21 1978
F

SMOOTH SURFACE APPROXIMATION BY A LOCAL METHOD OF INTERPOLATION AT SCATTERED POINTS

RICHARD FRANKE

Final Report for Period January - March 1978

Approved for Public Release; distributed unlimited

Prepared for: Chief of Naval Research
Arlington, VA 22217

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral Tyler F. Dedman
Superintendent

Jack Borsting
Provost

The work reported herein was supported by the Foundation Research Program of the Naval Postgraduate School with funds provided by the Chief of Naval Research.

This report was prepared by:

Richard Franke

RICHARD FRANKE
Associate Professor
Department of Mathematics

Reviewed by:

Carroll O. Wilde

CARROLL O. WILDE, Chairman
Department of Mathematics

Released by:

William M. Tolles

W. M. Tolles
(Acting) Dean of Research

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 14 NPS53-78-002	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER 9
4. TITLE (and Subtitle) 6 Smooth Surface Approximation by a Local Method of Interpolation at Scattered Points		5. TYPE OF REPORT & PERIOD COVERED Final rept. Jan - Mar 1978
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 Richard Franke		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Mathematics Naval Postgraduate School Monterey, CA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61152N, RR 000-01-01, ~~N0001478WR80023~~
11. CONTROLLING OFFICE NAME AND ADDRESS Chief of Naval Research Arlington, VA 22217		12. REPORT DATE 11 April 1978
		13. NUMBER OF PAGES 55 12 52p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 16 RR00001 17 RR0000101		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

scattered data, interpolation, surface approximation, optimal approximation, local methods

sub

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes a computer program which constructs a surface passing through a set of data points $(x_k, y_k, f_k), k = 1,\ldots,n$. It is based on previous work ~~of the author (JIMA 19 (1977) 471-482)~~, but uses a somewhat different approach which takes advantage of the nature of the approximations used and incorporates experience gained in the ensuing period. The surfaces are defined for all (x,y) points and have continuous second partial derivatives.

251 450

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

1. The Interpolation Problem

Given the set of data points (x_k, y_k, f_k), it is desired to construct a function $F(x,y)$, such that $F(x_k, y_k) = f_k, k = 1, \ldots, n$. For large sets of data it is desirable for the method to be local, that is, the value of $F(x,y)$ depends only on the value of f_k at nearby points (x_k, y_k).

This problem is receiving a great deal of attention and discussions of it and proposed methods can be found in [1], [2], [4], [7], and [8].

2.0. The Interpolation Scheme

This is a local method, the general idea having been discussed in [4]. The basic idea is to construct local interpolants, F_ℓ, which are then weighted by functions W_ℓ having limited support to obtain the function

$$F(x,y) = \sum_\ell W_\ell(x,y) F_\ell(x,y) / \sum_\ell W_\ell(x,y) \tag{1}$$

The details are fully discussed in the reference, but the important fact is that $F(x,y)$ will take on the value f_k at (x_k, y_k) if for each ℓ where $W_\ell(x_k, y_k) = \neq 0$, $F_\ell(x_k, y_k) = f_k$. In the referenced paper, the weight functions W_ℓ were taken to be of the form

$$W_\ell(x,y) = \begin{cases} 1 - 3(d_\ell/r_\ell)^2 + 2(d_\ell/r_\ell)^3 & , \; d_\ell \le r_\ell \\ 0 & , \; d_\ell > r_\ell \end{cases}$$

where r_ℓ is the radius of the smallest circle centered at (x_ℓ, y_ℓ) which contains a given fixed number of data points, and d_ℓ is the distance from (x,y) to (x_ℓ, y_ℓ).

This scheme, used with local interpolants, F_ℓ which were taken to be certain optimal approximations, yielded reasonably good results. However, the computational burden was rather high. This was due in part to a great deal of overlap in the regions where the W_ℓ are nonzero. In addition, the approxi-

mation is defined only on the union of the circles around each of the data points, which could cause problems.

One of the advantages of using optimal approximations is that the basis functions are generated by the (x_k, y_k) and the system is automatically nonsingular. Within some limitations, the resulting equations can easily be solved by Cholesky decomposition [5]. This opens the way for the present approach, which is to choose rectangular regions on which weight functions are non-zero, thus being able to carefully govern the amount of overlap with a resulting decrease in the necessary computations, as well as simplification of the weight functions.

2.1. Weight Functions

With these ideas in mind, we are now ready to describe the present selection of regions over which the weight functions are non-zero. These regions will be rectangles defined by the following parameters. Let n_x and n_y be given positive integers and let finite values of $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{n_x}$ and $\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_{n_y}$ be given. For convenience in this section we let $\tilde{x}_0 = \tilde{y}_0 = -\infty$ and $\tilde{x}_{n_x+1} = \tilde{y}_{n_y+1} = \infty$. For each $i = 1, \ldots, n_x$ and $j = 1, \ldots, n_y$, let R_{ij} denote the rectangle $(\tilde{x}_{i-1}, \tilde{x}_{i+1}) X (\tilde{y}_{j-1}, \tilde{y}_{j+1})$.

Let $H_5(s) = 1 - s^3(6s^2 - 15s + 10)$, the Hermite quintic satisfying $H_5(0) = 1$, $H_5'(0) = H_5''(0) = H_5(1) = H_5'(1) = H_5''(1) = 0$. We then define functions which are piecewise quintics with continuous second derivatives which have the property that they are non-zero on two intervals and satisfy

$$V_i(x_j) = \delta_{ij}, \; i = 1, \ldots, n_x, \; j = 0, 1, \ldots, n_x + 1$$

$$U_j(y_i) = \delta_{ji}, \; j = 1, \ldots, n_y, \; i = 0, 1, \ldots, n_y + 1 .$$

In particular,

$$V_1(x) = \begin{cases} 1 & , x < \tilde{x}_1 \\ H_5\left(\dfrac{x - \tilde{x}_1}{\tilde{x}_2 - \tilde{x}_1}\right) & , \tilde{x}_1 \le x < \tilde{x}_2 \\ 0 & , x \ge \tilde{x}_2 \end{cases}$$

$$V_i(x) = \begin{cases} 0 & , x < \tilde{x}_{i-1} \\ 1 - V_{i-1}(x) & , \tilde{x}_{i-1} \le x < \tilde{x}_i \\ H_5\left(\dfrac{x - \tilde{x}_i}{\tilde{x}_{i+1} - \tilde{x}_i}\right) & , \tilde{x}_i \le x < \tilde{x}_{i+1} \\ 0 & , x \ge \tilde{x}_{i+1} \end{cases}$$

for $i = 2,\ldots,n_x - 1$, and

$$V_{n_x}(x) = \begin{cases} 0 & , x < \tilde{x}_{n_x-1} \\ 1 - V_{n_x-1}(x) & \tilde{x}_{n_x-1} \le x < \tilde{x}_{n_x} \\ 1 & x \ge \tilde{x}_{n_x} . \end{cases}$$

The $U_j(Y)$ are dual. Then, if we define

$$W_{ij}(x,y) = V_i(x)U_j(Y),\ i = 1,\ldots,n_x,\ j = 1,\ldots,n_y ,$$

it is easily observed that the function $W_{ij}(x,y)$ has support $Cl(R_{ij})$ and that the functions form a partition of unity for the plane, i.e.,

$\sum_{i,j} W_{ij}(x,y) \equiv 1$ for all (x,y) .

These properties allow the construction of the interpolation function (1) to proceed easily since any point (x,y) is in at most four R_{ij} , and the denominator of (1) is always $\equiv 1$, allowing us to write

$$F(x,y) = \sum_{i,j} W_{ij}(x,y)Q_{ij}(x,y) .$$

We again emphasize that at most four terms in the sum are non-zero.

The appropriate choice of x_i and y_j as well as n_x and n_y depend on the data as well as the choice of local interpolating functions $Q_{ij}(x,y)$. For this reason we defer discussion of the selection of these grid lines until after we discuss the choice of $Q_{ij}(x,y)$.

2.2. Local Interpolation Functions

The only restriction on the local interpolation functions $Q_{ij}(x,y)$ are that they interpolate all data points in R_{ij} and that they are defined for all (x,y) in R_{ij}. Polynomials sometimes fail to satisfy these conditions. The use of optimal approximations in Sard corner spaces has been investigated [5], and for small numbers of data points, the approximations can be computed in straightforward fashion. One possible defect in such approximations is their lack of polynomial precision: even constants are not approximated exactly. With only a slight complication this can be overcome, since by a theorem of Barnhill and Gregory [3], the boolean sum operator $B \oplus L$ has the interpolation properties of B and the function precision of L. Here we are thinking of Bf as the optimal approximation in $B_{\lceil 2,2 \rceil}$ while Lf is the least squares fit by a linear function.

The implemented version of the program embodies three options: (1) Use optimal approximations in $B_{\lceil 2,2 \rceil}$ as the local interpolation functions; (2) Use the least squares linear approximation instead of an interpolation function; and (3) Use the optimal approximation in $B_{\lceil 2,2 \rceil}$ boolean sum the least squares linear approximation. The second option yields a surface which in general does not interpolate the given data. The third option is achieved computationally as $(B \oplus L)f = (B + L - BL)f = Lf + B(f - Lf)$.

The use of the boolean sum has a desirable effect in that it removes much of the effect of linear transformations of the data on the overall approximation.

However, for complete consistency with respect to translation and change of the measure of distance, each rectangle

$$[\tilde{x}_{i-1},\tilde{x}_{i+1}]X[\tilde{y}_{j-1},\tilde{y}_{j+1}]$$

is transformed to the unit square for the optimal approximation. For these purposes, we take $\tilde{x}_0 = \min_k x_k$ and $\tilde{x}_{n+1} = \max_k x_k$, and the dual in y . The base point (a,b) is taken to be $(0,0)$ for all $i,j \geq 2$, while it is taken to be $(1,1)$ for $i = j = 1$, $(0,1)$ for $i \geq 2, j = 1$, and $(1,0)$ for $i = 1, j \geq 2$. This yields lines of discontinuity in the second derivatives which are nowhere interior to the support regions for the weight functions W_{ij} , thus assuring continuous second derivatives in the overall approximation.

The overall approximation is invariant with respect to linear transformations which leave the directions of the axes unchanged. Since lines of discontinuities in the third derivatives occur along horizontal and vertical lines the approximation is not invariant with respect to rotations.

The points associated with R_{ij} include all the points in the closure of R_{ij} . Because approximation by a linear function requires at least three points, a parameter MINPTS , is used to assure that at least MINPTS points are selected for each R_{ij} . If extra points are required, they are taken as the closest points in the sup norm, distance being measured after

$$[x_{i-1},x_{i+1}]X[y_{j-1},y_{j+1}]$$

has been transformed onto $[0,1]^2$. Presently MINPTS is set to three and this has been satisfactory. It is easily changed, if desired or necessary. For example, if some R_{ij} has only three colinear points associated with it, the scheme will fail under options (2) or (3). Then one must either increase the value of MINPTS or use option (1).

2.3. Selection of Grid Lines

It is desirable to have automatic selection of grid lines, that is, values of $\tilde{x}_i$ and $\tilde{y}_j$. This should be accomplished in some manner which results in rectangles R_{ij} which contain approximately equal numbers of points. For data which is poorly distributed this may not be possible. However, for somewhat uniformly distributed points the process we describe here works quite well.

The selection of the grid lines is determined by one parameter, called NPPR, for "number of points per rectangle." The grid lines are then chosen so that there will be approximately NPPR points in each rectangle, R_{ij}. If there are additional points added to certain rectangles to make up MINPTS points the average may be higher. The average is, of course, dependent on the data set.

Equal numbers of grid lines are chosen in each direction, that is $n_x = n_y$. Because we want NPPR points per rectangle, each subrectangle

$$(\tilde{x}_i, \tilde{x}_{i+1}) X (\tilde{y}_j, \tilde{y}_{j+1})$$

should have $\frac{1}{4}$ NPPR points. Thus we want to choose $n_x = n_y$ so that $(n_x + 1)^2 \cdot \frac{1}{4}$ NPPR $= n$, the total number of data points. Thus, we take n_x to be the nearest integer to $(4n/\text{NPPR})^{1/2} - 1$.

Grid lines, that is $\tilde{x}_i$ and $\tilde{y}_j$ values, are now determined by choosing these values so that approximately $n/(n_x + 1)$ points occur in each $(\tilde{x}_i, \tilde{x}_{i+1}]$ and each $(\tilde{y}_j, \tilde{y}_{j+1}]$. Specifically, let $\hat{x}_k$ denote the values of x_k given in nondecreasing order, then $\tilde{x}_i = \hat{x}_k$, where k is the integer nearest $in/(n_x + 1)$ for $i = 1,2,\ldots,n_x$. The selection in y is dual.

3.0. Implementation

The scheme is implemented in a set of subprograms, only one of which is normally referenced by the user. The hierarchy of subprograms is given in figure 1. A brief description of them, according to level, follows.

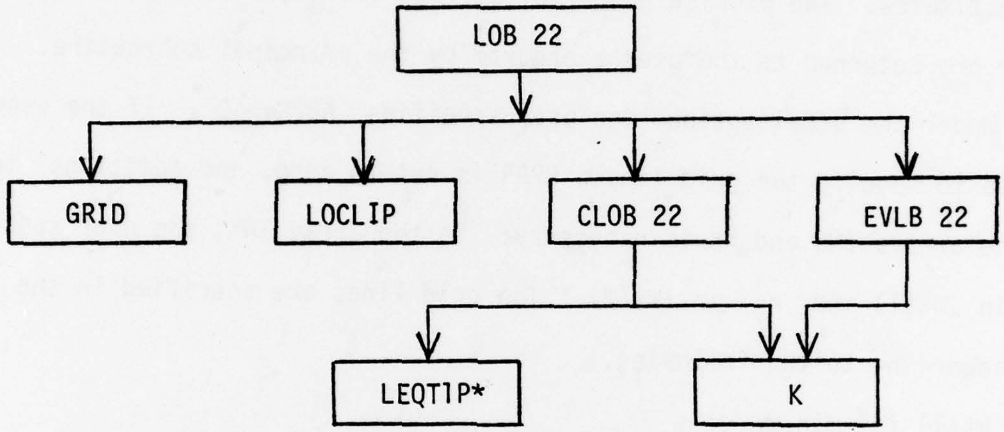


Figure 1

3.1.1. User's Program

The user's program must provide the data points, (x_k, y_k, f_k), $k = 1, \ldots, n$, as well as the points $x0_i$ and $y0_j$ for the grid of points at which the interpolation function is to be evaluated. In addition the user's program must provide workspace arrays, IWK and WK, and an array FO for the returned function values. The amount of storage required in the arrays IWK and WK are not known a priori, but are estimated as follows.

For IWK, approximately $4n(1 + 1/NPPR)$ locations are required. This has generally proven to be an overestimate. For WK, the storage required depends on the type of local approximation used and is approximately:

----
*This is a Cholesky decomposition equation solver in the IMSL Library, which itself references several other subroutines from the IMSL Library.

for MODE = 1, $4(n + \sqrt{n/NPPR})$;

for MODE = 2, $4(3n/NPPR + \sqrt{n/NPPR})$; and

for MODE = 3, $4(n + 3n/NPPR + \sqrt{n/NPPR})$.

As with the estimates for IWK, these estimates have proven to generally be overestimates. The precise number of storage locations required in each array are returned to the user's program by the principal subroutine.

Under the usual option, the user specifies NPPR > 0 . If the user wishes to specify the grid lines, NPPR is set to zero, and additional input in the arrays IWK and WK is necessary. In the array IWK, the user specifies n_x in IWK(1) and n_y in IWK(2). The grid lines are specified in the array WK, according to the following.

WK(1) is $\min_k x_k$,

WK(2),...,WK(n_x + 1) are the vertical grid values $\tilde{x}_1,..,\tilde{x}_{n_x}$, in increasing order,

WK(n_x + 2) is $\max_k x_k$,

WK(n_x + 3) is $\min_k y_k$,

WK(n_x + 4),...,WK(n_x + n_y + 3) are the horizontal grid values $\tilde{y}_1,...,\tilde{y}_{n_y}$ in increasing order, and

WK(n_x + n_y + 4) is $\max_k y_k$.

3.2.1. Subroutine LOB 22

This subroutine provides the interface between the user and the set of subroutines which implement the method. Generally LOB 22 sets up storage locations in the arrays IWK and WK, determines parameters required by other subroutines, and calls subroutines to (1) generate the grid (if necessary), (2) determine the interpolation points for each rectangle, (3) compute coefficients for the local interpolating functions, and finally, (4) to evaluate the interpolation function on the desired grid of points.

3.3.1. Subroutine GRID

This subroutine selects the values of $\tilde{x}_i$ and $\tilde{y}_j$ in accordance with the discussion in Section 2.3.

3.3.2. Subroutine LOCLIP

This subroutine determines the local interpolation points for each R_{ij} in accordance with the last paragraph of Section 2.2.

3.3.3. Subroutine CLOB 22

This subroutine computes the coefficients in the least squares plane (MODE = 2 or 3) and the coefficients in the optimal approximation (MODE = 1 or 3) or each of the rectangles R_{ij} . In the present implementation the IMSL Cholesky decomposition equation solver LEQT1P is used. This could be replaced at a facility where IMSL is not available, although according to IMSL policy, LEPT1P (and associated subroutines) can be used as part of this package at any facility. Because of the short single precision word length on the IBM 360/370 series computers, on which this program was implemented, the coefficients for the system of equations for the optimal approximation are generated in double precision. On computers with a longer word length, the double precision variables in this routine can be safely removed. Other double precision statements occur in EVLB 22 and function K, which must also be removed.

3.3.4. Subroutine EVLB 22

This subroutine evaluates the approximation (2) on the set of points $(x0_i, y0_j)$ as specified by the user, and returns the values in F0. As noted above, the double precision variables in this subroutine should be removed on computers with longer word length than the IBM 360/370 series.

3.4.1. Function K

This function evaluates the representers for point evaluation functionals in $B_{[2,2]}$. For evaluation at (u,v) , base point at (a,b) , the representer, as a function of (s,t) [6] is

$$K(a,b;u,v,s,t) = g_2(a;u,s)g_2(b;v,t) \text{ , where}$$

$$g_2(a;u,s) = G_2(a;u,s) = (s-u)_+^{(3)} + 1 + (u-a)(s-a) + (u-a)(s-a)_+^{(2)} + (s-a)_+^{(3)}$$

$$\text{for } a \le u \text{ , and}$$

$$g_2(a;u,s) = G_2(-a;-u,-s) \text{ for } u < a \text{ .}$$

The arguments of this function are all single precision, but because of the short word length of the IBM 360/370 computers, all calculations are performed in double precision, and the returned value is double precision. On computers with longer word lengths these calculations can be done in single precision.

4.0. Examples

The method has been applied to a number of sets of data with good results. Figures 3 - 5 show test surfaces and results of applying the method for each of the options for local approximations. The three surfaces are described by

(C) $F(x,y) = \tanh(y - x) + 1$,

(S) $F(x,y) = 3/2[\cos(3/5(y - 1)) + 5/4]/[1 + (\frac{x-4}{3})^2]$, and

(E) $$F(x,y) = 9\{3/4 \exp(\frac{-(x-3)^2 - (y-3)^2}{4}) + \exp(-(\frac{x}{7})^2 - (\frac{y}{10}))$$
$$- \frac{1}{5}\exp(-(x - 5)^2 - (y - 8)^2)$$
$$+ \frac{1}{2}\exp(\frac{-(x-8)^2 - (y-4)^2}{4})\} ,$$

respectively. The 100 interpolation points were chosen at random within a unit square centered at (i,j) for $i,j = 1,2,...,10$. The points are shown in Figure 2 as +'s, with the convex hull shown by dashed lines, while the square

$(1,10)^2$, on which the resulting interpolation functions were evaluated, is given by the solid lines. The diagonal line shows the direction toward the viewing point.

There does not appear to be a great deal of difference between the optimal approximation and the optimal approximation boolean sum least squares plane. Generally the latter option has slightly smaller errors and slightly less noticeable defects. Gross defects in the approximations can generally be traced to a lack of data in that particular part of the region.

The effect of varying the parameter NPPR is shown in Figures 6 - 14. Some general observations are possible from this set of views. Most apparent is the fact that option (2), the least squares plane fit as the local approximation does not appear to lead to very good results. In general, however, the smaller value of NPPR gives better results, visually, and usually better accuracy, too.

The choice of NPPR = 6 for options (1) and (3) appears to be a reasonable one. For surfaces with sharp gradients, as in Figure 6, it appears that localizing the behavior as much as possible with a smaller value of NPPR is the best strategy. For smooth surfaces, such as in Figure 9 and 12 it appears the opposite is true, where NPPR = 8 seems to lead to the best results.

The storage and timing results are given in Table 1. The storage refers to requirements of the two workspace arrays provided by the user. The timing is for calculation of the 1089 points generated for the plots. The program was run under the Fortran H compiler on the IBM 360 model 67 at the Naval Postgraduate School. Computation times are dependent on external factors and may vary from run to run.

5.0. Acknowledgements

During the first half of 1977 the author was a Visiting Associate Professor at the University of Utah. Interactions with Professor R. E. Barnhill and his students on the subject of surface approximation proved to be fruitful. The kernel of a number of ideas in the present scheme germinated during that time. Thanks also go to Rosemary E. Chang of Sandia Laboratories (Livermore) who first undertook to run the program on a CDC computer. Improvements in the program description and the test program were a result of those efforts.

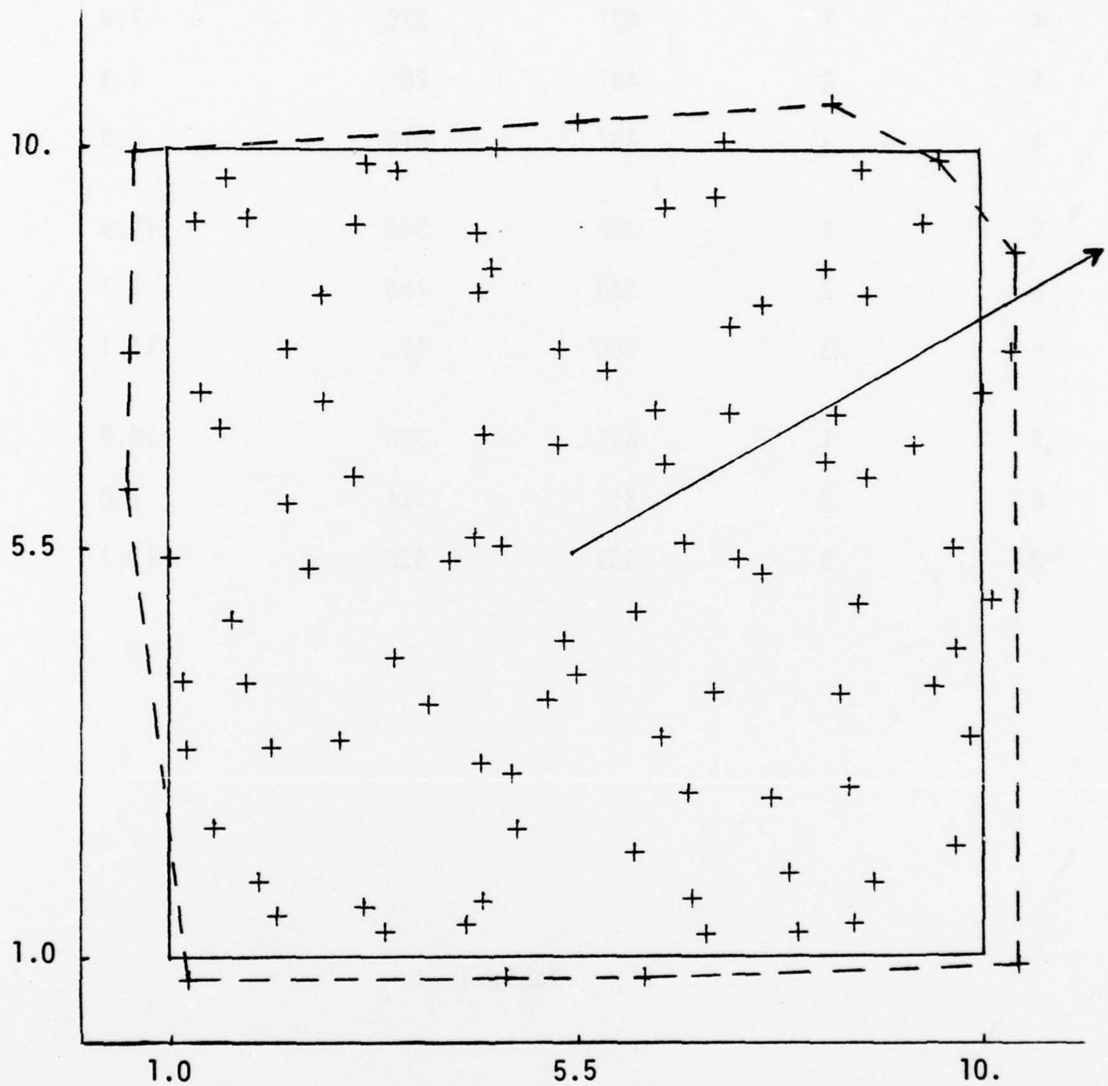
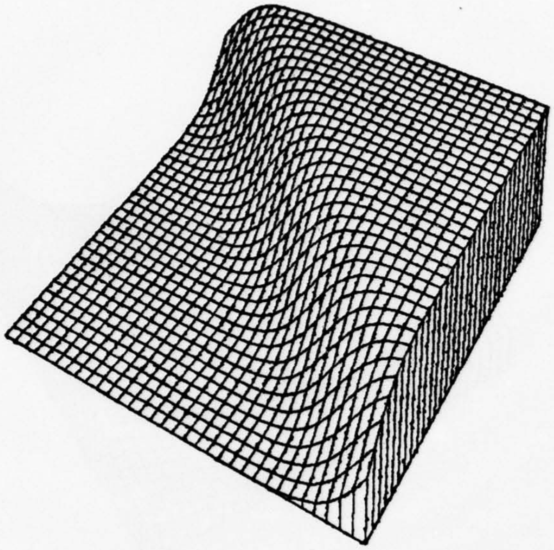


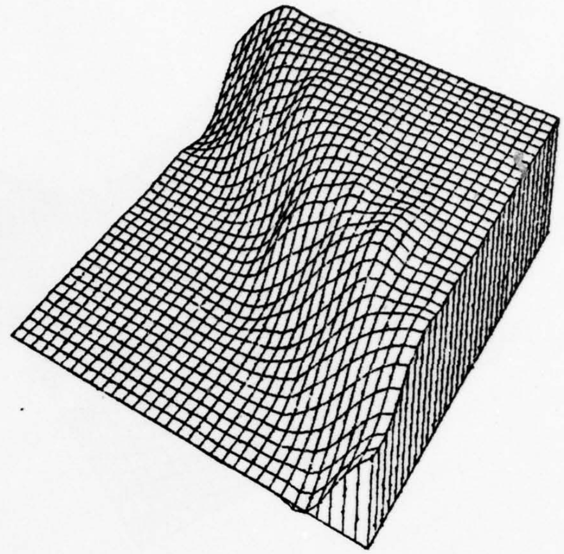
Figure 2

NPPR	MODE	NIWK	NWK	Approx. Time (sec.)
4	1	437	375	7.9
4	2	437	265	1.3
4	3	437	618	8.2
6	1	380	346	10.9
6	2	380	165	1.1
6	3	380	493	11.1
8	1	351	328	12.9
8	2	351	124	1.0
8	3	351	436	13.1

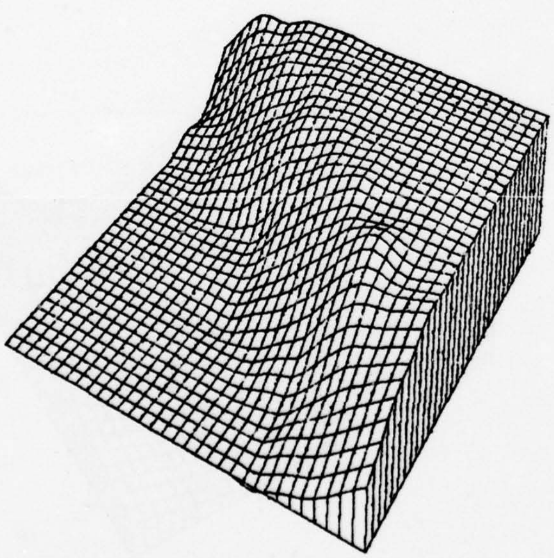
Table 1



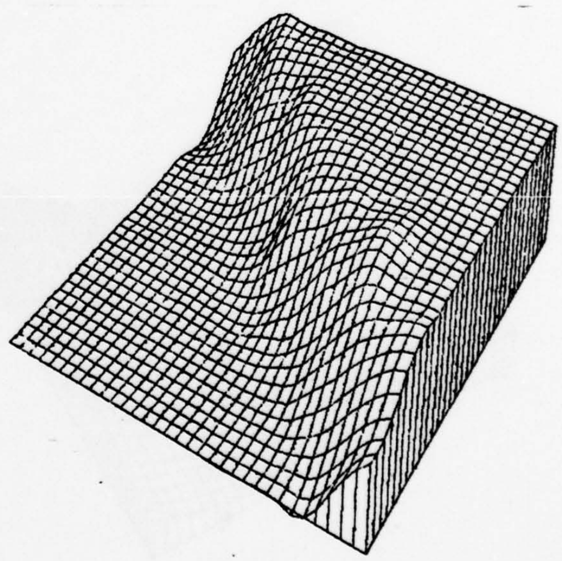
Test Surface
Cliff Function



Mode = 1 , E_{max} = .468
E_{rms} = .0263
E_{mean} = .0526

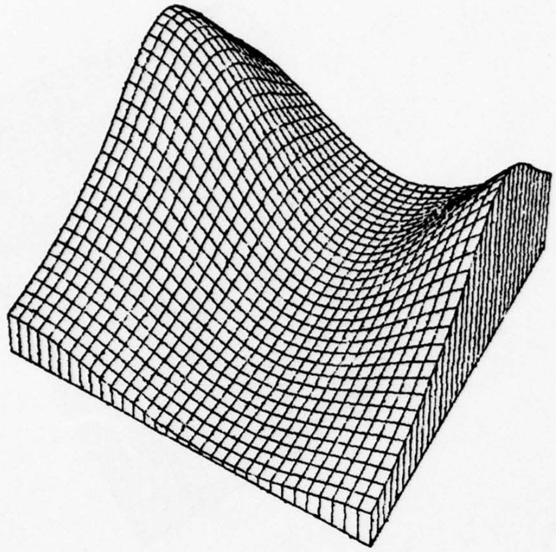


Mode = 2 , E_{max} = .283
E_{rms} = .0523
E_{mean} = .0864



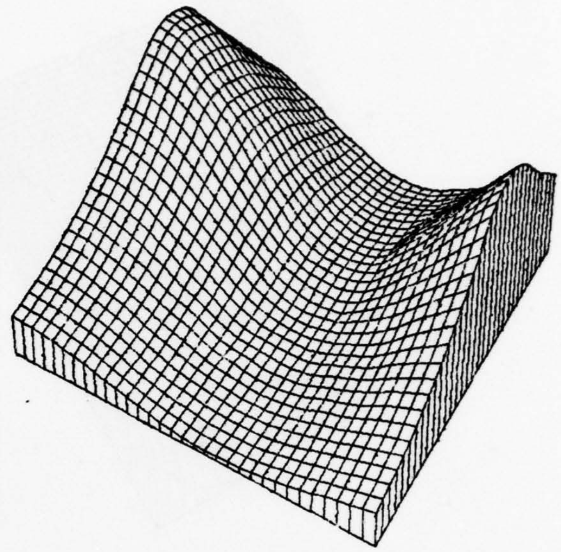
Mode = 3 , E_{max} = .466
E_{rms} = .0257
E_{mean} = .0527

Figure 3 (NPPR = 6)

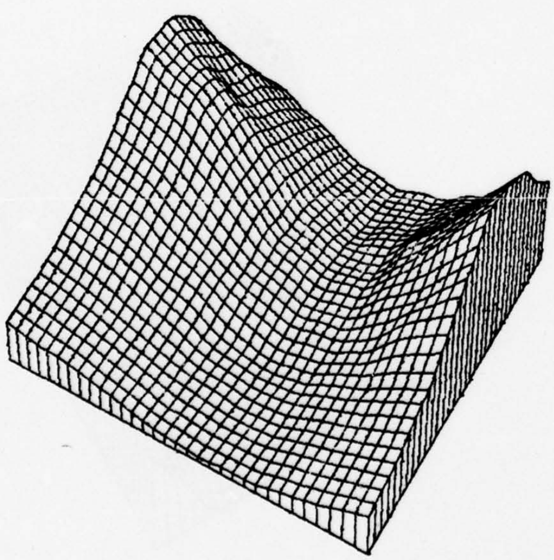


Test Surface

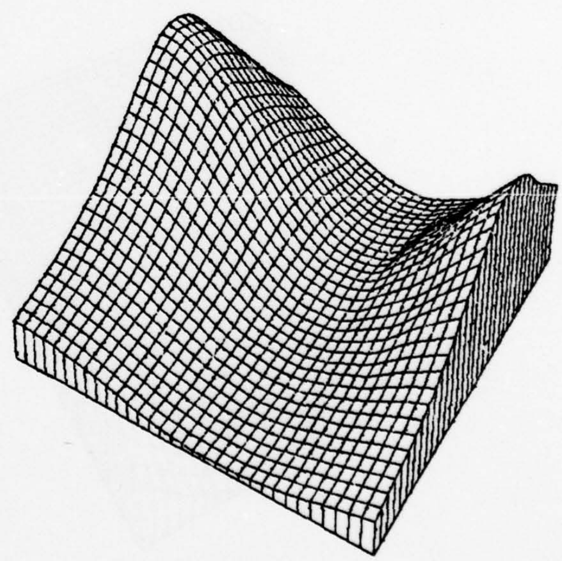
Saddle Function



Mode = 1 , E_{max} = .187
E_{rms} = .0156
E_{mean} = .0273

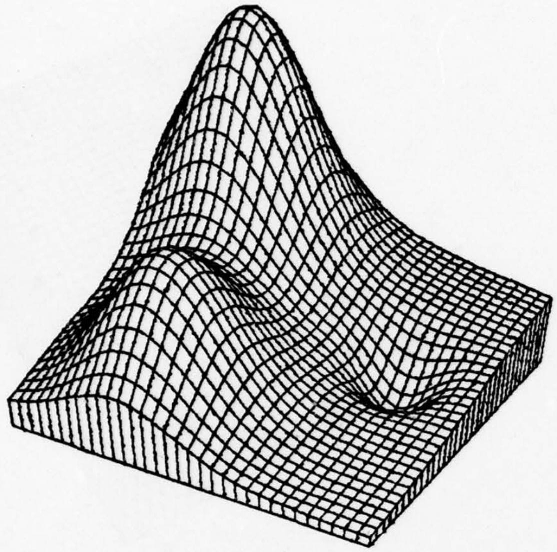


Mode = 2 , E_{max} = .389
E_{rms} = .0495
E_{mean} = .0739



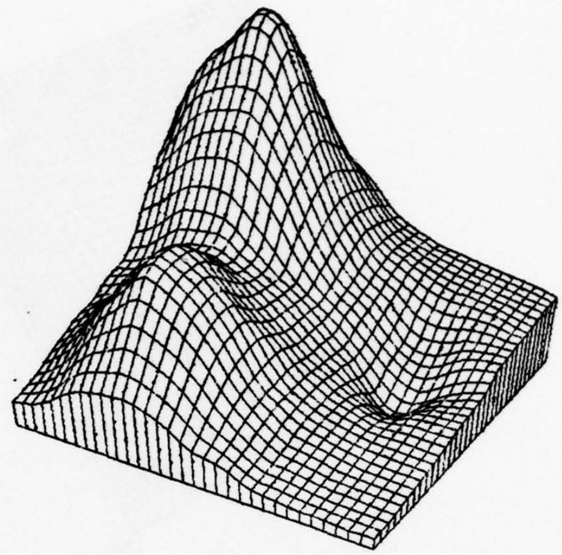
Mode = 3 , E_{max} = .178
E_{rms} = .0148
E_{mean} = .0265

Figure 4 (NPPR = 6)

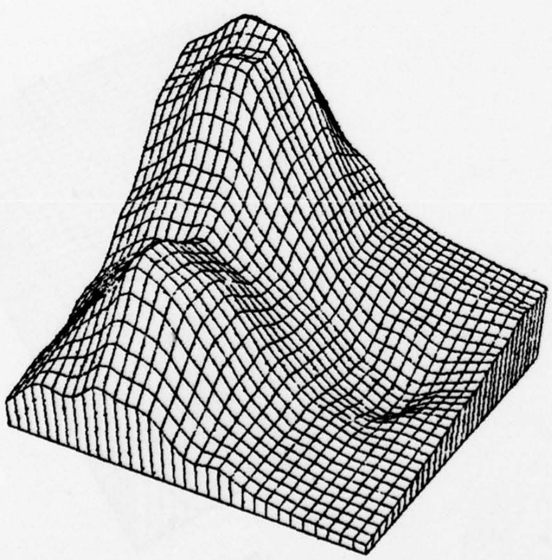


Test Surface

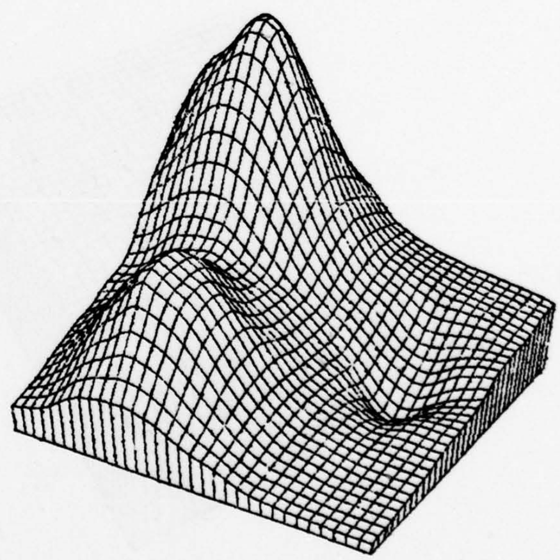
Exponentials



Mode = 1 , E_{max} = .974
E_{rms} = .0929
E_{mean} = .169

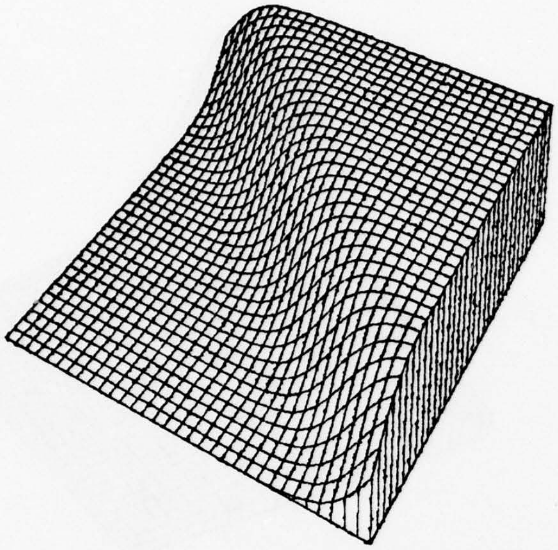


Mode = 2 , E_{max} = .216
E_{rms} = .209
E_{mean} = .366



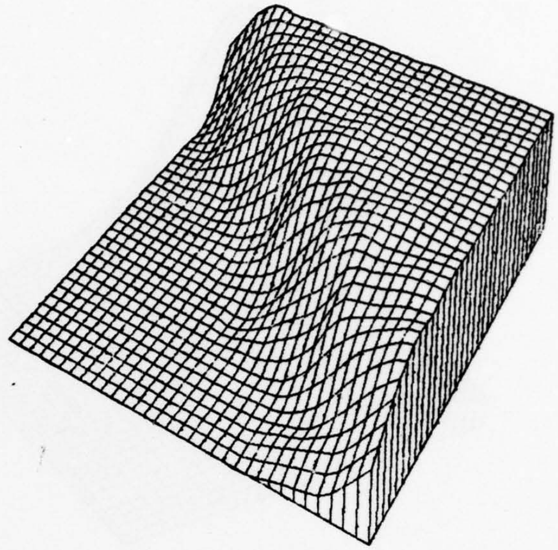
Mode = 3 , E_{max} = .827
E_{rms} = .0757
E_{mean} = .133

Figure 5 (NPPR = 6)

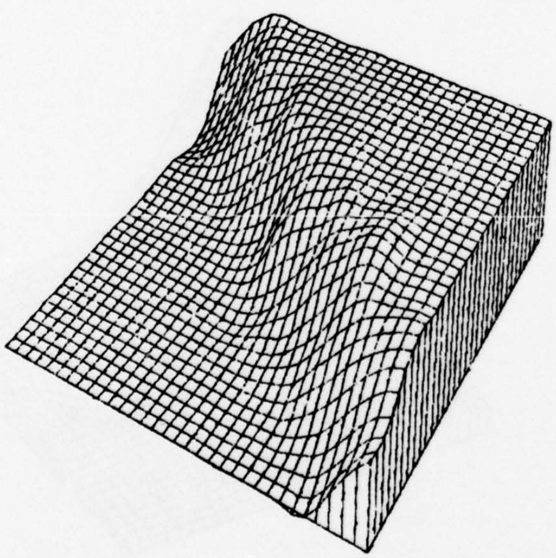


Test Surface

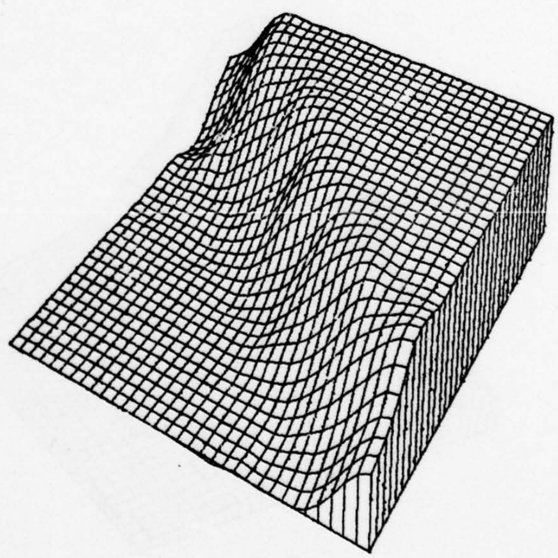
Cliff Function



NPPR = 4 , E_{max} = .265
E_{rms} = .0271
E_{mean} = .0513

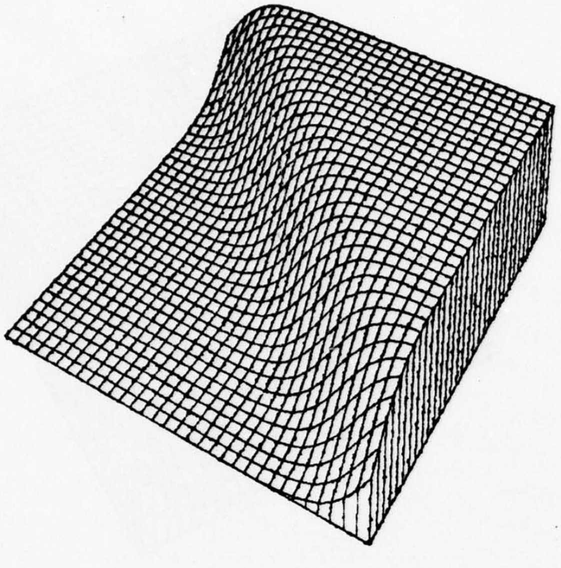


NPPR = 6 , E_{max} = .466
E_{rms} = .0257
E_{mean} = .0527



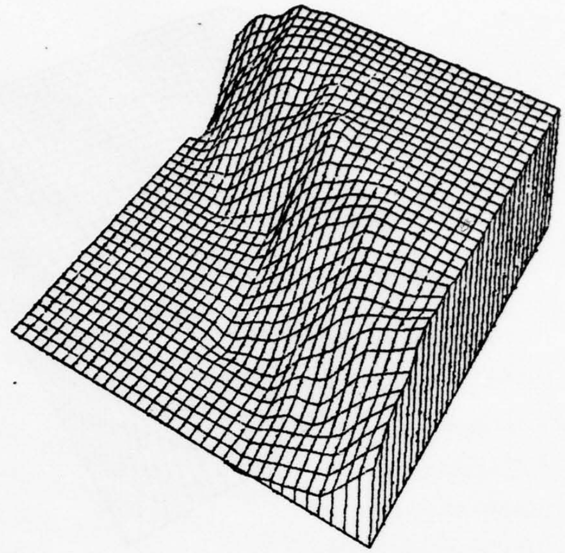
NPPR = 8 , E_{max} = .467
E_{rms} = .0308
E_{mean} = .0633

Figure 6 (Mode = 1)

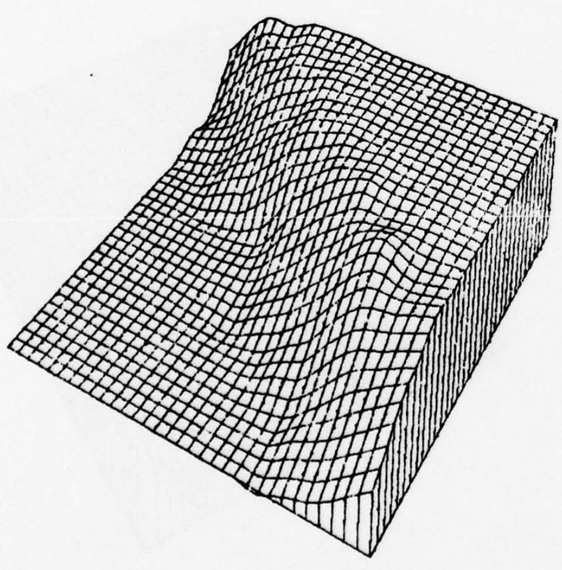


Test Surface

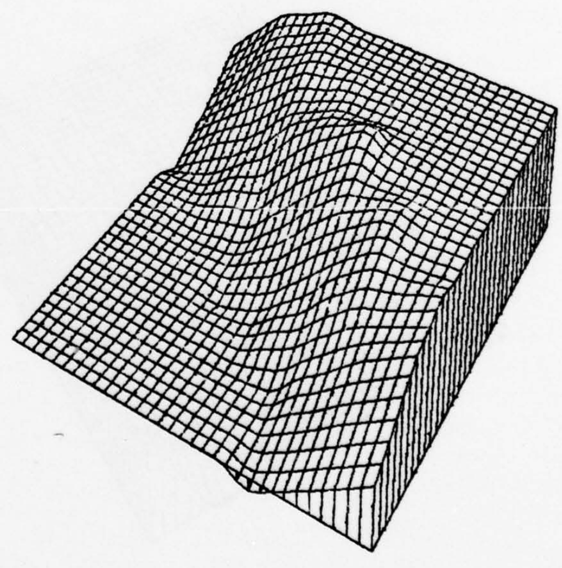
Cliff Function



NPPR = 4 , E_{max} = .336
E_{rms} = .0435
E_{mean} = .0797

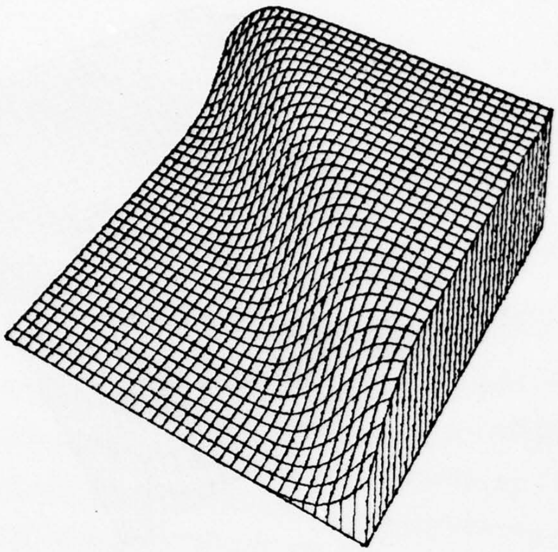


NPPR = 6 , E_{max} = .283
E_{rms} = .0523
E_{mean} = .0864

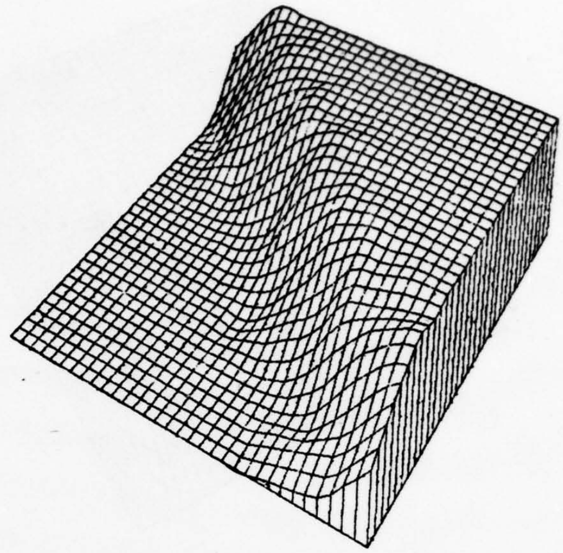


NPPR = 8 , E_{max} = .375
E_{rms} = .0692
E_{mean} = .113

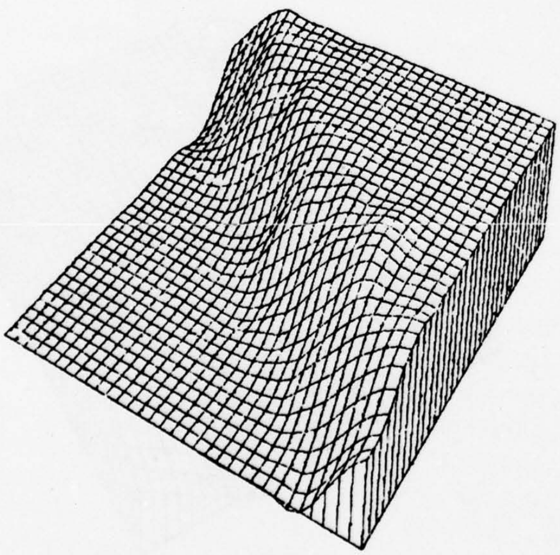
Figure 7 (Mode = 2)



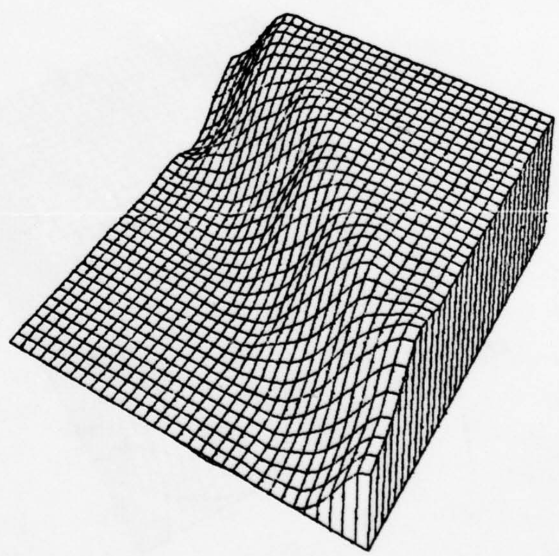
Test Surface
Cliff Function



NPPR = 4 , E_{max} = .261
E_{rms} = .0246
E_{mean} = .0500

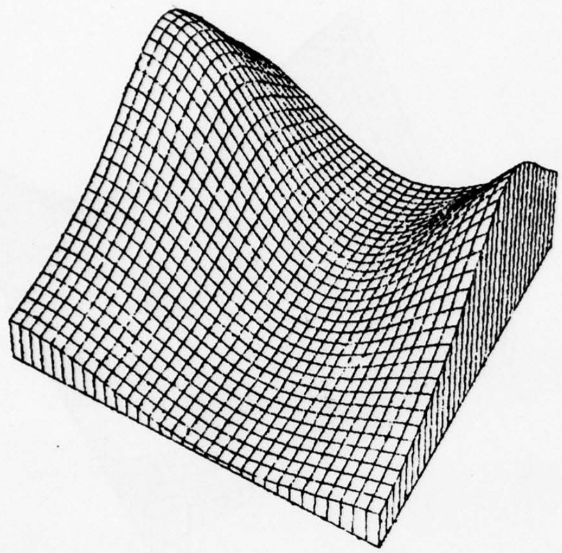


NPPR = 6, E_{max} = .468
E_{rms} = .0263
E_{mean} = .0526



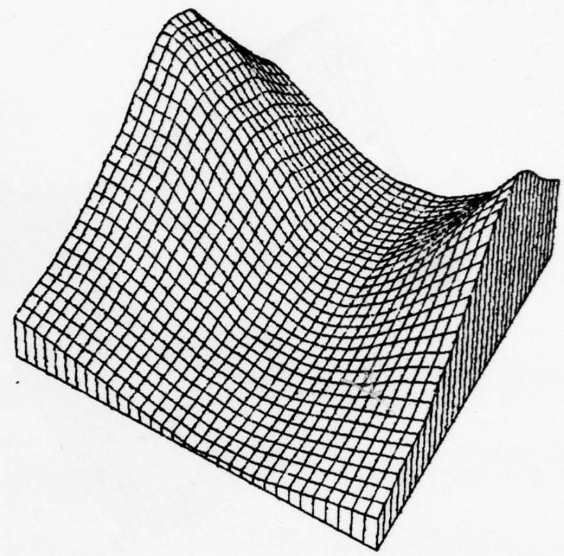
NPPR = 8 , E_{max} = .462
E_{rms} = .0304
E_{mean} = .0622

Figure 8 (Mode = 3)

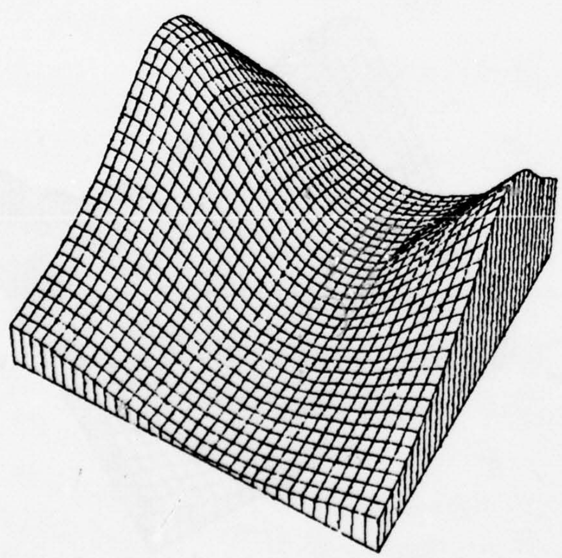


Test Surface

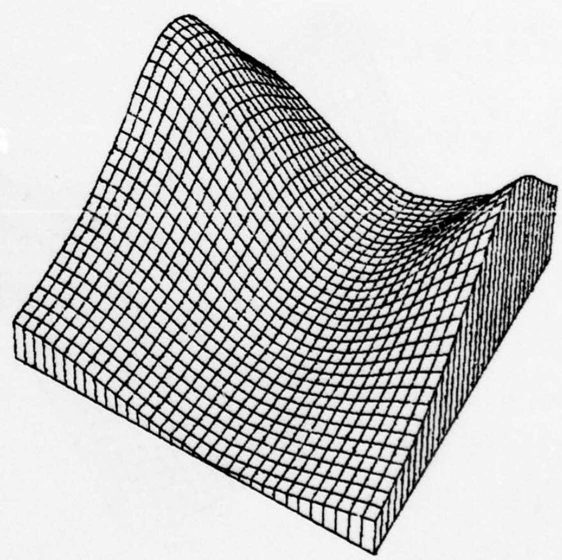
Saddle Function



NPPR = 4 , E_{max} = .208
E_{rms} = .0249
E_{mean} = .0398

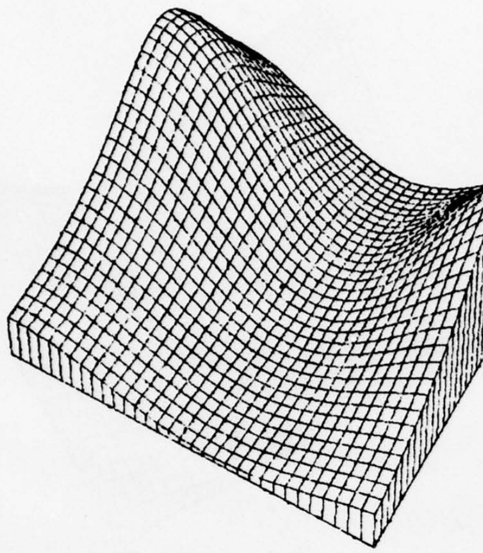


NPPR = 6 , E_{max} = .187
E_{rms} = .0156
E_{mean} = .0273

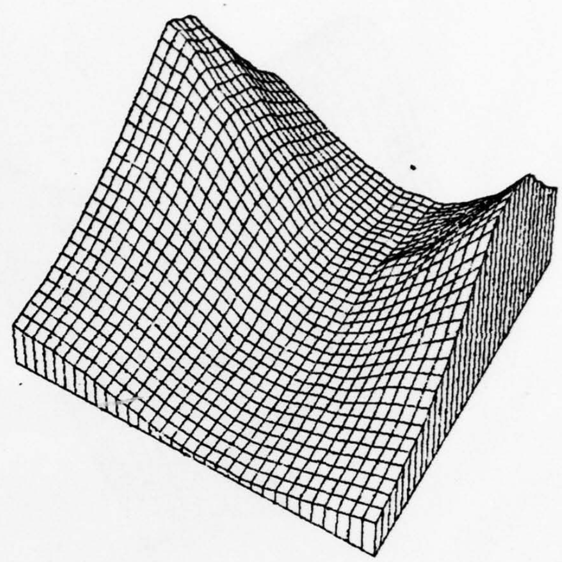


NPPR = 8 , E_{max} = .154
E_{rms} = .0118
E_{mean} = .0211

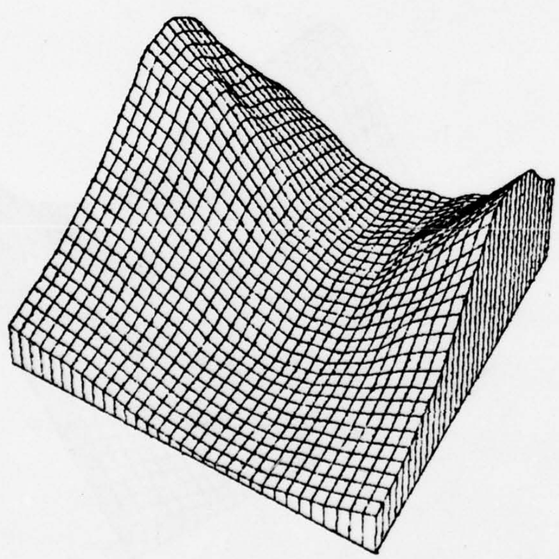
Figure 9 (Mode = 1)



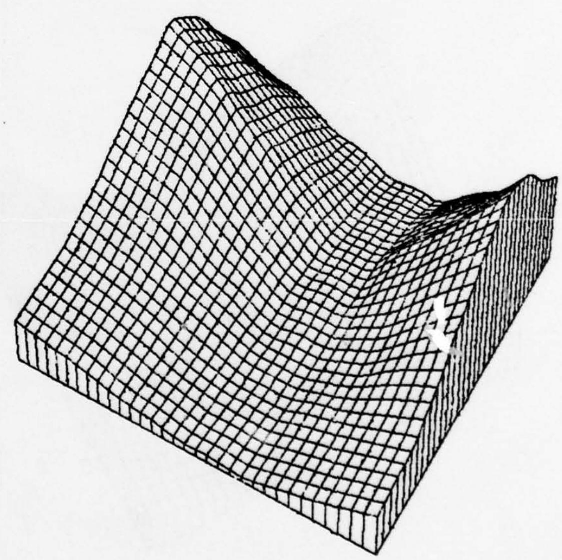
Test Surface
Saddle Function.



NPPR = 4 , E_{max} = .247
E_{rms} = .0338
E_{mean} = .0487

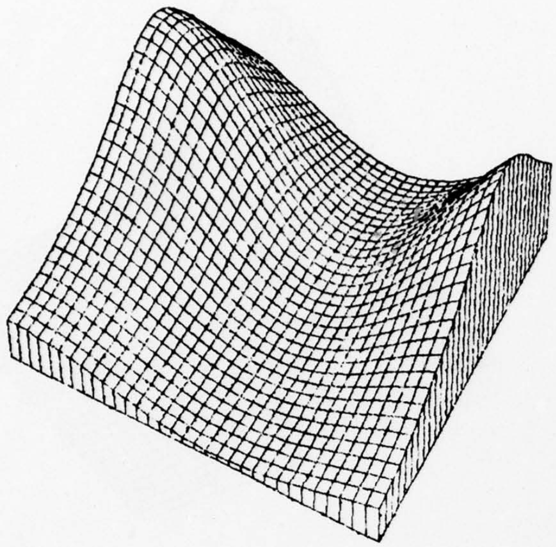


NPPR = 6 , E_{max} = .389
E_{rms} = .0495
E_{mean} = .0739



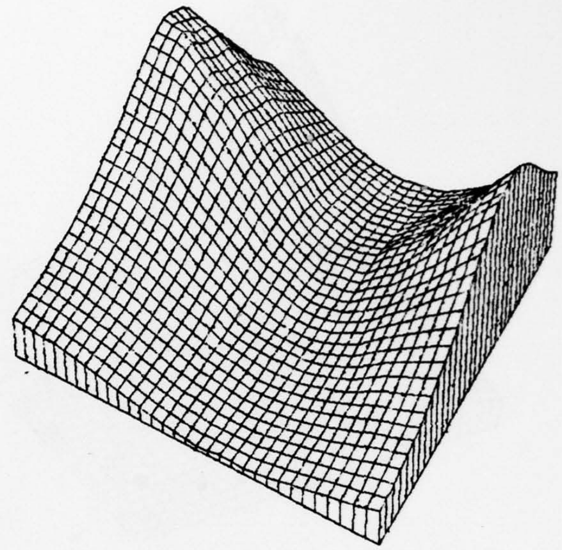
NPPR = 8 , E_{max} = .336
E_{rms} = .0565
E_{mean} = .0803

Figure 10 (Mode = 2)

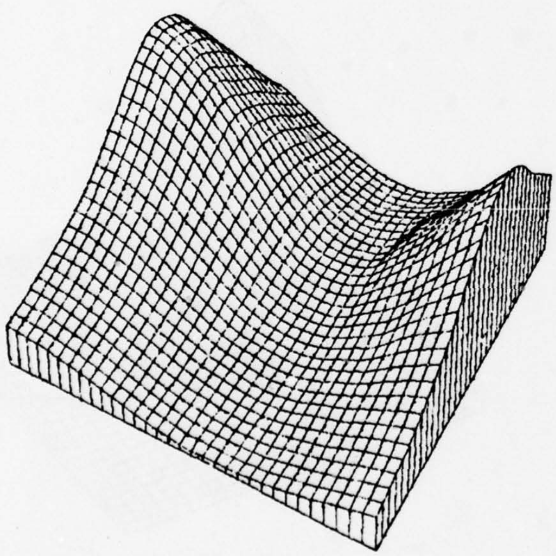


Test Surface

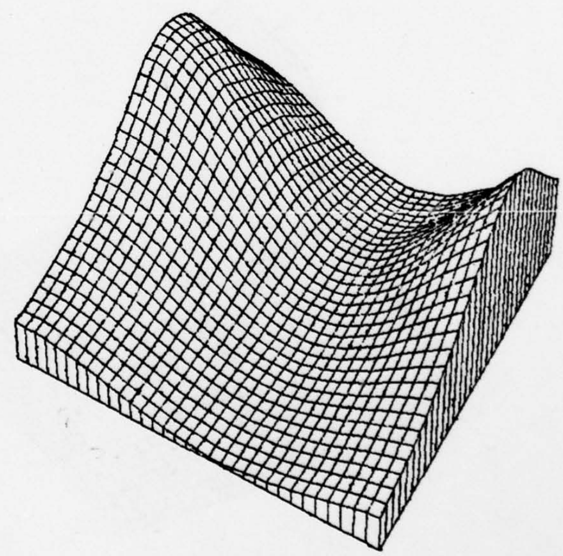
Saddle Function



NPPR = 4 , E_{max} = .244
E_{rms} = .0211
E_{mean} = .0363

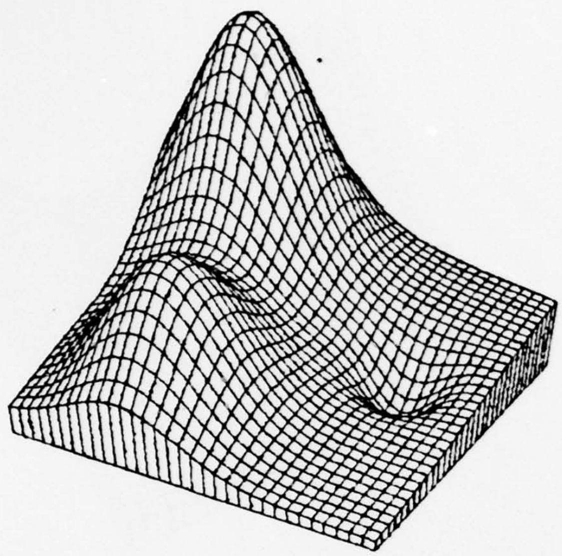


NPPR = 6 , E_{max} = .178
E_{rms} = .0148
E_{mean} = .0265

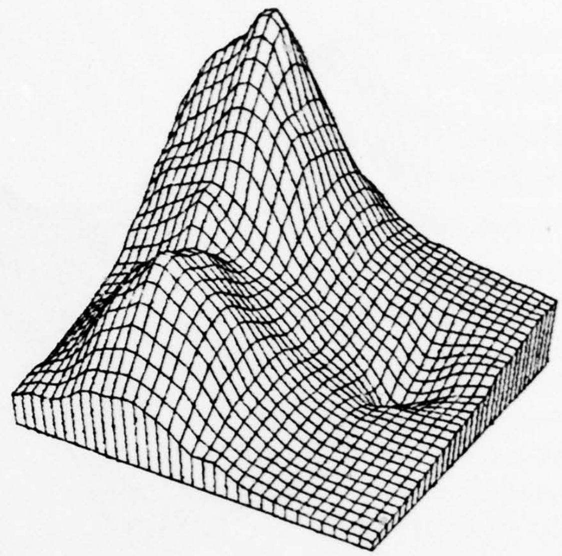


NPPR = 8 , E_{max} = .148
E_{rms} = .0115
E_{mean} = .0202

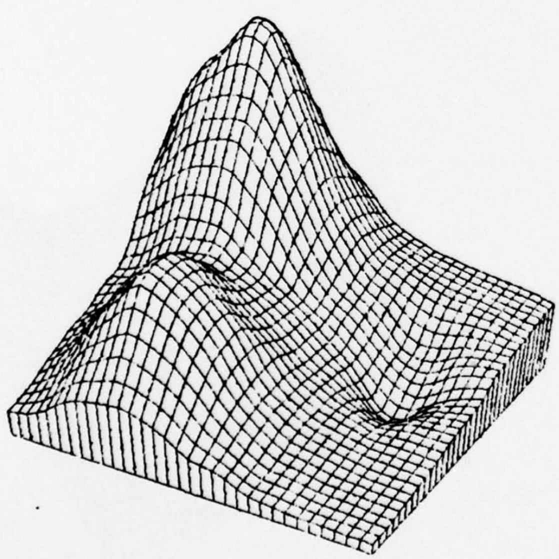
Figure 11 (Mode = 3)



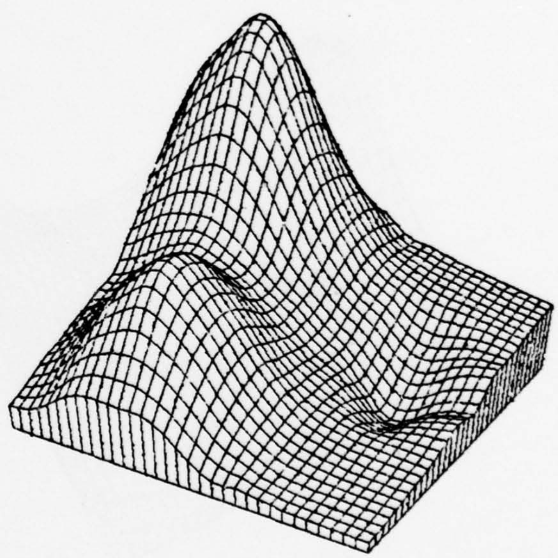
Test Surface
Exponentials



NPPR = 4 , E_{max} = 1.20
E_{rms} = .128
E_{mean} = .227

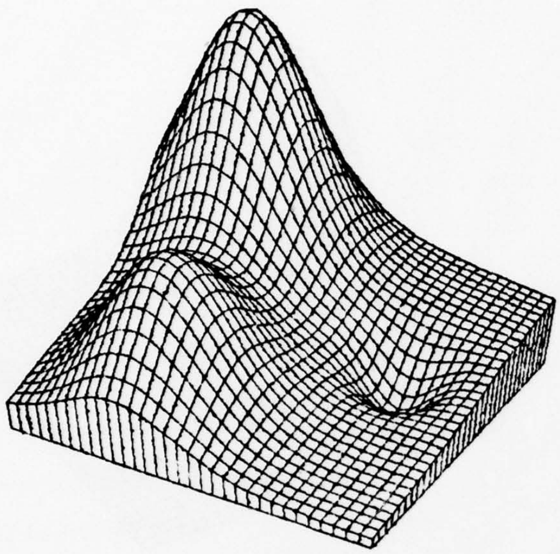


NPPR = 6 , E_{max} = .974
E_{rms} = .0929
E_{mean} = .169



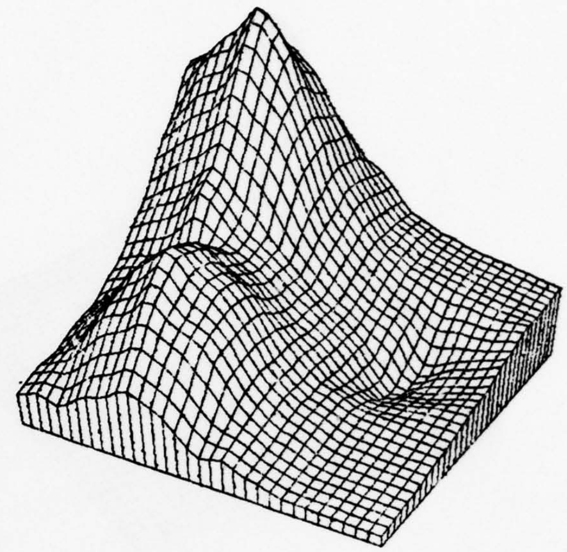
NPPR = 8 , E_{max} = .703
E_{rms} = .0779
E_{mean} = .129

Figure 12 (Mode = 1)



Test Surface

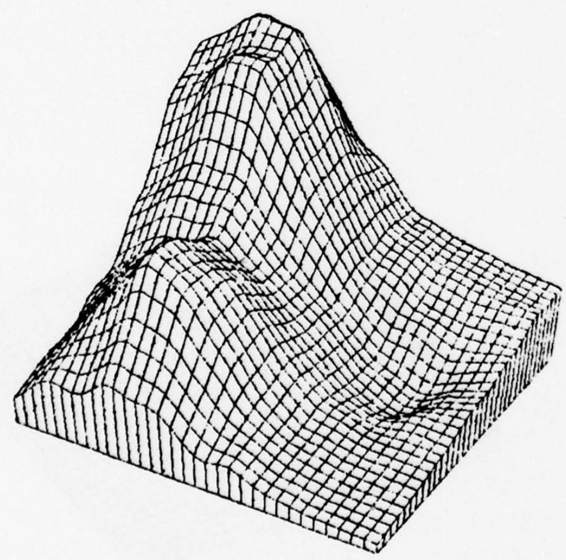
Exponentials



NPPR = 4 , E_{max} = 1.29

E_{rms} = .162

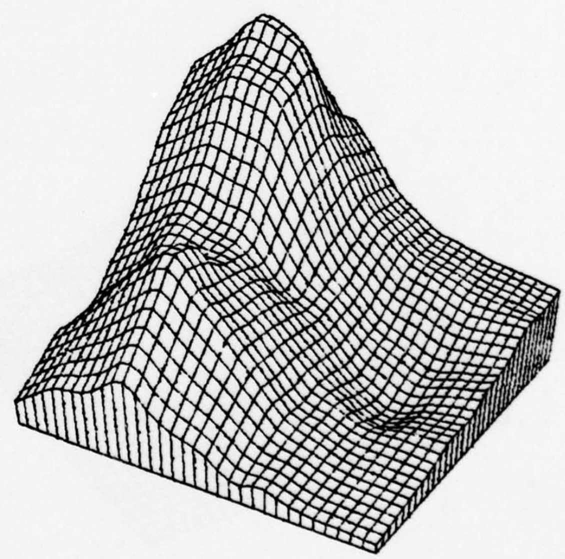
E_{mean} = .265



NPPR = 6 , E_{max} = 2.16

E_{rms} = .209

E_{mean} = .366

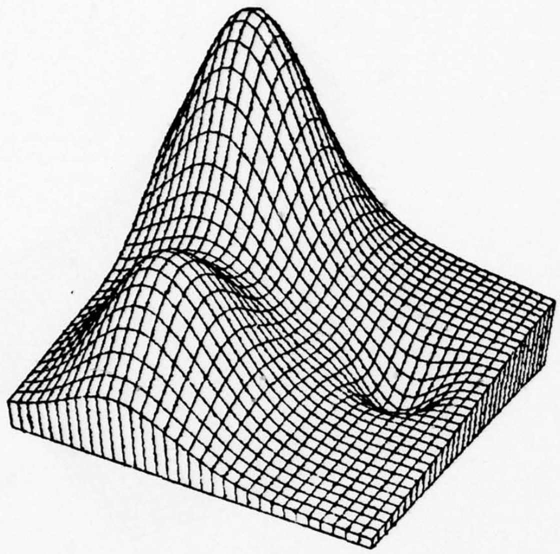


NPPR = 8 , E_{max} = 2.04

E_{rms} = .251

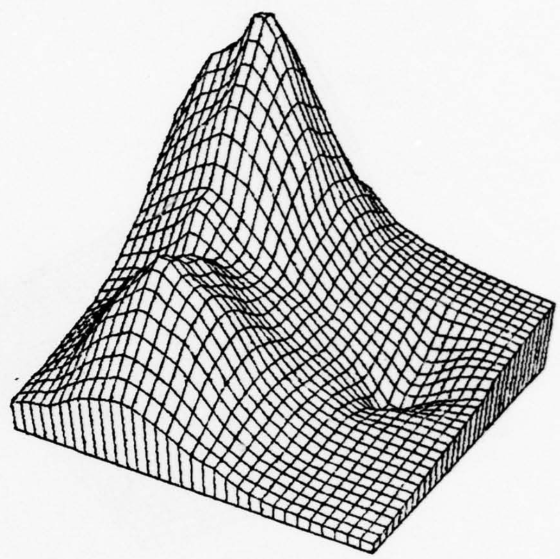
E_{mean} = .398

Figure 13 (Mode = 2)

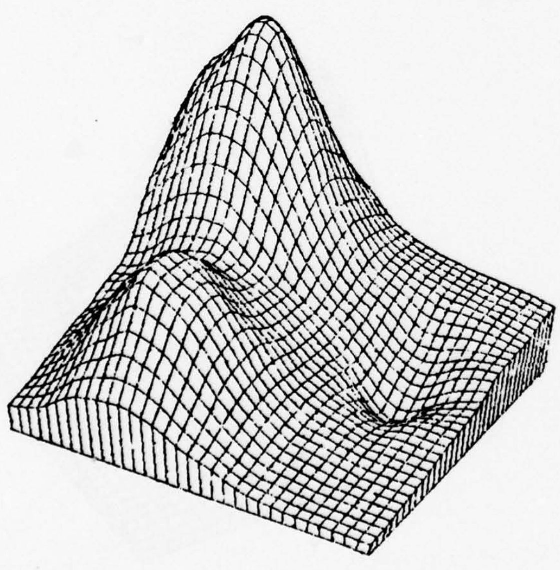


Test Surface

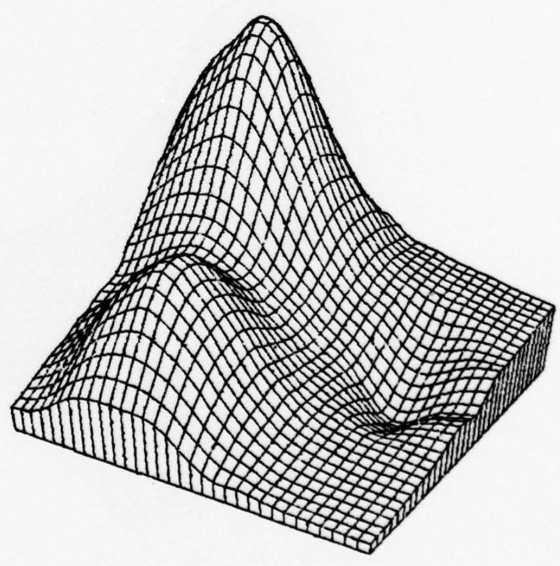
Exponentials



NPPR = 4 , E_{max} = 1.30
E_{rms} = .101
E_{mean} = .189



NPPR = 6 , E_{max} = .827
E_{rms} = .0757
E_{mean} = .133



NPPR = 8 , E_{max} = .748
E_{rms} = .0778
E_{mean} = .131

Figure 14 (Mode = 3)

Appendix: Program Listings and Sample Output

```
      DIMENSION X(25), Y(25), F(25), XO(3), YO(3), FO(3,3), FAC(3,3,4),  TST00010
     1 E(3,3), IWK(100), WK(120)                                          TST00020
      DATA NOUT/6/                                                        TST00030
      DATA FAC / 2.557525042 , 1.156302903 , 0.343660482 ,                TST00040
     1           1.686594904 , 0.729563464 , 0.211667614 ,                TST00050
     2           0.737497040 , 0.343550908 , 0.093714050 ,                TST00060
     3           2.658681228 , 1.522739473 , 0.682157263 ,                TST00070
     4           1.754474583 , 0.877377736 , 0.284611789 ,                TST00080
     5           0.933759918 , 0.413208351 , 0.109657470 ,                TST00090
     6           2.572645288 , 1.158049264 , 0.348002564 ,                TST00100
     7           1.694989367 , 0.724123412 , 0.214020225 ,                TST00110
     8           0.736691672 , 0.341271688 , 0.093442275 ,                TST00120
     9           2.376248312 , 1.107179709 , 0.348220775 ,                TST00130
     A           1.626663927 , 0.747442886 , 0.217782524 ,                TST00140
     B           0.711956600 , 0.324808684 , 0.097759180 /                TST00150
      K = 0                                                               TST00160
C                                                                         TST00170
      DO 100 I=1,5                                                        TST00180
      DO 100 J=1,5                                                        TST00190
      K = K+1                                                             TST00200
      X(K) = I-1+1./FLOAT(K)                                              TST00210
      Y(K) = J-1./FLOAT(K)                                                TST00220
  100 F(K) = 4.*EXP(-(X(K)**2+Y(K)**2)*.2)                                TST00230
C                                                                         TST00240
      NP = 25                                                             TST00250
C                                                                         TST00260
      DO 120 I=1,3                                                        TST00270
      XO(I) = I+.5                                                        TST00280
  120 YO(I) = I-.5                                                        TST00290
C                                                                         TST00300
      DO 160 MODE=1,3                                                     TST00310
      NWK = 120                                                           TST00320
      NIWK = 100                                                          TST00330
      CALL LOB22 (MODE,6,NP,X,Y,F,3,XO,3,YO,IWK,NIWK,WK,NWK,FO,KER)       TST00340
C                                                                         TST00350
      DO 140 I=1,3                                                        TST00360
      DO 140 J=1,3                                                        TST00370
  140 E(I,J) = FAC(I,J,MODE)-FO(I,J)                                      TST00380
C                                                                         TST00390
      WRITE (NOUT,2) MODE,KER,NIWK,NWK                                    TST00400
  160 WRITE (NOUT,1) FO,E                                                 TST00410
C                                                                         TST00420
      IWK(1) = 2                                                          TST00430
      IWK(2) = 3                                                          TST00440
      WK(1) = .2                                                          TST00450
      WK(2) = 1.5                                                         TST00460
      WK(3) = 3.5                                                         TST00470
      WK(4) = 4.+1./21.                                                   TST00480
```

```
      WK(5) = 0.                                                        TST00490
      WK(6) = 1.25                                                      TST00500
      WK(7) = 2.9                                                       TST00510
      WK(8) = 3.8                                                       TST00520
      WK(9) = 4.96                                                      TST00530
      NWK = 120                                                         TST00540
      NIWK = 100                                                        TST00550
      MODE = 3                                                          TST00560
      CALL LOB22 (MODE,O,NP,X,Y,F,3,XO,3,YO,IWK,NIWK,WK,NWK,FO,KER)     TST00570
C                                                                       TST00580
      DO 180 I=1,3                                                      TST00590
      DO 180 J=1,3                                                      TST00600
  180 E(I,J) = FO(I,J)-FAC(I,J,4)                                       TST00610
C                                                                       TST00620
      WRITE (NOUT,2) MODE,KER,NIWK,NWK                                  TST00630
      WRITE (NOUT,1) FO,E                                               TST00640
      STOP                                                              TST00650
C                                                                       TST00660
    1 FORMAT (/7X,15HFUNCTION VALUES,3(/3F20.6)//7X,50HDEVIATIONS (THESETST00670
     1 VALUES REPRESENT ROUNDOFF ERROR)/(3F20.6))                       TST00680
    2 FORMAT (43H0THE VALUES OF MODE, KER, NIWK, AND NWK ARE,4I5)       TST00690
      END                                                               TST00700
```

```
      SUBROUTINE LOB22 (MODE,NPPR,NPI,XI,YI,FI,NXC,XO,NYO,YO,
     1                  IWK,NIWK,WK,NWK,FO,KER)
C
C     THIS SUBROUTINE SERVES AS A USER INTERFACE TO THE SET OF
C     SUBROUTINES THAT IMPLEMENT FRANKE'S METHOD CF SURFACE INTERPO-
C     LATION.  RECTANGLULAR REGIONS ARE USED WITH PRODUCT QUINTIC
C     HERMITE WEIGHT FUNCTIONS.  THE RECTANGLES ARE CHOSEN IN AN ATTEMPT
C     TO OBTAIN ABOUT NPPR POINTS IN EACH REGION.  THE SAME NUMBER OF
C     GRID LINES IS USED IN EACH DIRECTION.  LOCAL INTERPOLATION FUNC-
C     TIONS ARE EITHER OPTIMAL APPROXIMATIONS IN SARD SPACE B CORNER
C     2,2 OR OPTIMAL APPROXIMATIONS IN B CORNER 2,2 BOOLEAN SUM THE
C     LEAST SQUARES PLANE, OR FOR APPROXIMATION RATHER THAN INTERPO-
C     LATION, THE LEAST SQUARES PLANE.
C
C     THE ARGUMENTS ARE AS FOLLOWS.
C
C        MODE - INPUT.  INDICATES THE STATUS OF THE CALCULATION.
C                 = 1,  SET UP THE PROBLEM.  USE CPTIMAL APPROXIMATIONS
C                       FOR THE LOCAL APPROXIMATICNS, AND RETURN THE
C                       THE GRID OF INTERPOLATED POINTS INDICATED BY
C                       NXO, XO, NYO, YC, IN FO.
C                 = 2,  SET UP THE PROBLEM.  USE THE LEAST SQUARES
C                       PLANE FOR THE LCCAL APPROXIMATION, AND RETURN
C                       THE GRID OF INTERPOLATED POINTS INDICATED BY
C                       NXO, XO, NYO, YC IN FO.
C                 = 3,  SET UP THE PROBLEM.  USE THE OPTIMAL APPROXI-
C                       MATIONS BOOLEAN SUM THE LEAST SQUARES PLANE FOR
C                       THE LOCAL APPROXIMATIONS, AND RETURN THE GRID
C                       OF INTERPOLATED POINTS INDICATED BY NXC, XO,
C                       NYO, YO IN FO.
C                 = 4,  THE PROBLEM HAS BEEN SET UP PREVIOUSLY.  CALCU-
C                       LATE THE GRID OF INTERPOLATED POINTS INDICATED
C                       BY NXO, XO, NYO, YO IN FO.  THE PRCGRAM ASSUMES
C                       THAT ALL ARGUMENTS EXCEPT NXO, XO, NYO, YO, FO,
C                       AND MODE ARE UNCHANGED FRCM THE PREVIOUS CALL.
C        NPPR - INPUT.  DESIRED AVERAGE NUMBER OF POINTS PER REGION.
C                       THE SUGGESTED VALUE IS SIX.  SHOULD BE AT LEAST
C                       FOUR.  VALUES LARGER THAN TEN ARE NOT ADVISED.
C                       IF THE USER WISHES TO SPECIFY HIS CWN GRID LINES
C                       X TILDA AND Y TILDA, HE MAY DO SC BY SETTING
C                       NPPR = 0 AND SETTING NECESSARY VALUES IN THE
C                       ARRAYS IWK AND WK.
C        NPI  - INPUT.  NUMBER OF INPUT DATA POINTS.
C        XI   -
C        YI   - INPUT.  THE DATA POINTS (XI,YI,FI), I=1,...,NPI.
C        FI   -
C        NXO  - INPUT.  THE NUMBER OF XO VALUES AT WHICH THE INTERP-
C                       OLATION FUNCTION IS TO BE CALCULATED.
```

```
C      XO    - INPUT.   THE VALUES OF X AT WHICH THE INTERPOLATION        LOB   490
C                       FUNCTION IS TO BE CALCULATED.                     LOB   500
C      NYO   - INPUT.   THE NUMBER OF YO VALUES AT WHICH THE INTERP-      LOB   510
C                       OLATION FUNCTION IS TO BE CALCULATED.             LOB   520
C      YO    - INPUT.   THE VALUES OF Y AT WHICH THE INTERPOLATION        LOB   530
C                       FUNCTION IS TO BE CALCULATED.                     LOB   540
C      IWK   - INPUT AND OUTPUT.  THIS ARRAY IS OUTPUT WHEN MODE = 1, 2LOB   550
C                       OR 3, AND IS INPUT WHEN MODE = 4.  THIS MUST BE LOB   560
C                       AN ARRAY DIMENSIONED APPROXIMATELY 4*NPI*         LOB   570
C                       (1 + 1/NPPR).  FOR NPPR = 6, THIS MEANS ABOUT     LOB   580
C                       5*NPI.  WHEN NPPR IS INPUT AS ZERO THE USER MUSTLOB   590
C                       SPECIFY THE NUMBER OF VERTICAL GRID LINES (THE    LOB   600
C                       NUMBER OF X TILDA VALUES) IN IWK(1) AND THE       LOB   610
C                       NUMBER OF HORIZONTAL GRID LINES (THE NUMBER OF    LOB   620
C                       Y TILDA VALUES) IN IWK(2).                        LOB   630
C      NIWK  - INPUT AND OUTPUT.  WHEN MODE = 1, 2, OR 3 THIS MUST BE    LOB   640
C                       SET TO THE NUMBER OF LOCATIONS RESERVED FOR THE LOB   650
C                       ARRAY IWK.  ON OUTPUT IT IS SET TO THE ACTUAL     LOB   660
C                       REQUIREMENTS FOR THE ARRAY.  IF THE ARRAY IWK     LOB   670
C                       IS NOT DIMENSIONED LARGE ENOUGH IT MAY CAUSE      LOB   680
C                       A FAILURE SINCE THE ARRAY BOUNDS WILL BE          LOB   690
C                       EXCEEDED IN SUBROUTINE LOCLIP.                    LOB   700
C      WK    - INPUT AND OUTPUT.  THIS ARRAY IS OUTPUT WHEN MODE = 1, 2LOB   710
C                       OR 3, AND IS INPUT WHEN MODE = 4.  THIS MUST BE LOB   720
C                       AN ARRAY DIMENSIONED APPROXIMATELY AS FOLLOWS.    LOB   730
C                         FOR MODE=1, 4*(NPI+SQRT(NPI/NPPR)).             LOB   740
C                         FOR MODE=2, 4*(3*NPI/NPPR+SQRT(NPI/NPPR)).      LOB   750
C                         FOR MODE=3, 4*(NPI+3*NPI/NPPR+SQRT(NPI/NPPR)).LOB   760
C                       FOR NPPR = 6, THIS MEANS APPROXIMATELY            LOB   770
C                         FOR MODE=1, 4*NPI + 1.6*SQRT(NPI)               LOB   780
C                         FOR MODE=2, 2*NPI + 1.6*SQRT(NPI)               LOB   790
C                         FOR MODE=3, 6*NPI + 1.6*SQRT(NPI)               LOB   800
C                       WHEN NPPR IS INPUT AS ZERO THE USER MUST SPECIFYLOB   810
C                       THE VALUES OF X TILDA AND Y TILDA AS FOLLOWS.     LOB   820
C                       WK(1) IS MIN(XI), WK(2),...,WK(NXG+1) ARE THE     LOB   830
C                       NXG (= IWK(1)) VALUES OF X TILDA IN INCREASING    LOB   840
C                       ORDER, AND WK(NXG+2) IS MAX(XI).  WK(NXG+3) IS    LOB   850
C                       MIN(YI), WK(NXG+4),...,WK(NXG+NYG+3) ARE THE NYGLOB   860
C                       (= IWK(2)) VALUES OF Y TILDA IN INCREASING ORDERLOB   870
C                       AND WK(NXG+NYG+4) IS MAX(YI).                     LOB   880
C      NWK   - INPUT AND OUTPUT.  ON ENTRY WITH MODE = 1, 2, OR 3, THISLOB   890
C                       MUST BE THE NUMBER OF LOCATIONS RESERVED FOR THELOB   900
C                       ARRAY WK.  ON OUTPUT IT IS SET TO THE ACTUAL      LOB   910
C                       REQUIREMENTS FOR THE ARRAY.                       LOB   920
C      FO    - OUTPUT.  VALUES OF THE INTERPOLATION FUNCTION AT THE      LOB   930
C                       GRID OF POINTS INDICATED BY NXO, XO, NYO, YO.     LOB   940
C                       FO IS ASSUMED TO BE DIMENSIONED (NXO,NYO) IN THELOB   950
C                       CALLING PROGRAM.                                  LOB   960
```

```
C         KER   - OUTPUT.  RETURN INDICATOR.                            LOB  970
C                     = 0,  NORMAL RETURN.                              LOB  980
C                     =-1,  PROBLEM HAS NOT BEEN PREVIOUSLY SET UP (LOB22 LOB  990
C                           CALLED WITH MODE = 1, 2 OR 3)               LOB 1000
C                     = 1,  ERROR RETURN FROM CLOB22, SINGULAR MATRIX IN THELOB 1010
C                           LEAST SQUARES FIT.                          LOB 1020
C                     = 2,  ERROR RETURN FROM CLOB22, SINGULAR MATRIX IN THELOB 1030
C                           OPTIMAL FIT.                                LOB 1040
C                     = 3,  ERROR RETURN FROM CLOB22.  SOME RECTANGLE  (I,J)LOB 1050
C                           HAS MORE THAN THE ALLOWED NUMBER OF POINTS  LOB 1060
C                           ASSOCIATED WITH IT.                         LOB 1070
C                           SUBROUTINE CLOB22.                          LOB 1080
C                     = 4,  PREVIOUS ERROR RETURN FROM CLOB22 HAS NOT BEEN LOB 1090
C                           CORRECTED.                                  LOB 1100
C                     = 5,  IWK AND WK ARRAYS HAVE NOT BEEN DIMENSIONED  LOB 1110
C                           LARGE ENOUGH IN THE CALLING PROGRAM.  REDIMEN- LOB 1120
C                           SION IWK AND WK TO AT LEAST THE SIZE INDICATED LOB 1130
C                           BY NIWK AND NWK, RESPECTIVELY.              LOB 1140
C                     = 6,  MODE IS OUT OF RANGE.                       LOB 1150
C                                                                       LOB 1160
      DIMENSION XI(1), YI(1), FI(1), IWK(1), WK(1), XO(1), YO(1), FO(NXOLOB 1170
     1,1)                                                               LOB 1180
      DATA KERO/-1/                                                     LOB 1190
      IF (MODE.LT.1.OR.MODE.GT.4) GO TO 220                             LOB 1200
      KER = 0                                                           LOB 1210
C                                                                       LOB 1220
C     ON INITIAL ENTRY MODE = 1, 2, OR 3, AND THE GRID LINES ARE SET UP,LOB 1230
C     LOCAL INTERPOLATION POINTS ARE DETERMINED AND LOCAL APPROXIMATIONSLOB 1240
C     ARE COMPUTED.                                                     LOB 1250
C                                                                       LOB 1260
      IF (MODE.EQ.4) GO TO 140                                          LOB 1270
      NWKIN = NWK                                                       LOB 1280
      NIWKIN = NIWK                                                     LOB 1290
      NXGWK = 1                                                         LOB 1300
      NPWK = 3                                                          LOB 1310
      IF (NPPR.LE.0) GO TO 100                                          LOB 1320
      NXG = SQRT(4.*FLOAT(NPI)/FLOAT(NPPR))-.5                          LOB 1330
      NYG = NXG                                                         LOB 1340
      IWK(1) = NXG                                                      LOB 1350
      IWK(2) = NYG                                                      LOB 1360
      GO TO 120                                                         LOB 1370
  100 NXG = IWK(1)                                                      LOB 1380
      NYG = IWK(2)                                                      LOB 1390
  120 IALWK = NXG+NYG+5                                                 LOB 1400
      IABWK = IALWK                                                     LOB 1410
      IF (MODE.NE.1) IABWK=IABWK+3*NXG*NYG                              LOB 1420
      NYGWK = NXG+3                                                     LOB 1430
      MPWK = NXG*NYG+4                                                  LOB 1440
```

```
      IF (NPPR.GT.0) CALL GRID (XI,YI,NPI,NXG,WK(NXGWK),NYG,WK(NYGWK),WK
     1(IALWK))
C
C     DETERMINE THE LOCAL INTERPOLATICN POINTS FOR THE REGIONS.
C
      CALL LOCLIP (NXG,WK(NXGWK),NYG,WK(NYGWK),NPI,XI,YI,IWK(NPWK),IWK(M
     1PWK),WK(IALWK))
      NWK = IABWK-1
      IF (MODE.NE.2) NWK = NWK+IWK(MPWK-1)-1
      NIWK = NXG*NYG+2+IWK(MPWK-1)
      IF (NIWK.GT.NIWKIN) GO TO 200
      IF (NWK.GT.NWKIN) GO TO 200
      MC = MODE
C
C     COMPUTE THE LOCAL APPROXIMATIONS.
C
      CALL CLOB22 (MO,XI,YI,FI,NXG,WK(NXGWK),NYG,WK(NYGWK),IWK(NPWK),IWK
     1(MPWK),WK(IALWK),WK(IABWK),IER)
      KERO = IER
      IF (IER.NE.0) GO TO 160
  140 IF (KERO.NE.0) GO TO 180
C
C     COMPUTE THE FUNCTION VALUES ON THE DESIRED GRID OF POINTS.
C
      CALL EVLB22 (MO,XI,YI,NXG,WK(NXGWK),NYG,WK(NYGWK),IWK(NPWK),IWK(MP
     1WK),WK(IALWK),WK(IABWK),NXO,XO,NYC,YO,FO)
      RETURN
C
C     ERROR RETURNS
C
  160 KER = IER
      RETURN
  180 KER = 4
      IF (KERO.LT.0) KER = -1
      RETURN
  200 KER = 5
      RETURN
  220 KER = 6
      RETURN
      END
```

```
      SUBROUTINE GRID (X,Y,N,NX,XG,NY,YG,T)
C     THIS SUBROUTINE PLACES A SET OF GRID LINES CN THE SET OF POINTS
C     (X,Y), I=1,...,N .  THIS IS DONE BY PLACING APPROXIMATELY EQUAL
C     NUMBERS OF THEM WITHIN VERTICAL ANC HORIZONTAL LINES.
C
C     THE ARGUMENTS ARE AS FOLLOWS.
C
C       X,Y  - INPUT.  THE ARRAY OF (X,Y) POINTS.
C       N    - INPUT.  THE NUMBER OF (X,Y) POINTS.
C       NX   - INPUT.  THE DESIRED NUMBER OF VERTICAL GRID LINES.
C       XG   - OUTPUT.  THE COORDINATES OF THE VERTICAL GRID LINES.
C       NY   - INPUT.  THE DESIRED NUMBER OF HORIZONTAL GRID LINES.
C       YG   - OUTPUT.  THE COORDINATES OF THE HORIZONTAL GRID LINES.A
C       T    - WORK ARRAY OF DIMENSION AT LEAST N.
C
      DIMENSION X(1), Y(1), XG(1), YG(1), T(1)
      NM1 = N-1
C
      DO 100 I=1,N
  100 T(I) = X(I)
C
      K = 1
      FINC = FLOAT(N)/FLOAT(NX+1)
      GO TO 220
C
  120 DC 140 J=1,NX
      K = J*FINC+.5
  140 XG(J+1) = T(K)
C
      XG(1) = T(1)
      XG(NX+2) = T(N)
C
      DO 160 I=1,N
  160 T(I) = Y(I)
C
      K = 2
      FINC = FLOAT(N)/FLOAT(NY+1)
      GO TO 220
C
  180 DO 200 J=1,NY
      K = J*FINC+.5
  200 YG(J+1) = T(K)
C
      YG(1) = T(1)
      YG(NY+2) = T(N)
      RETURN
C
  220 DC 260 I=1,NM1
```

```
      IP1 = I+1                                                          GRI  490
C                                                                        GRI  500
      DO 240 J=IP1,N                                                     GRI  510
      IF (T(I).LE.T(J)) GO TO 240                                        GRI  520
      TS = T(I)                                                          GRI  530
      T(I) = T(J)                                                        GRI  540
      T(J) = TS                                                          GRI  550
  240 CONTINUE                                                           GRI  560
C                                                                        GRI  570
  260 CONTINUE                                                           GRI  580
C                                                                        GRI  590
      GO TO (120,180), K                                                 GRI  600
      END                                                                GRI  610
```

```
      SUBROUTINE LOCLIP (NXG,XG,NYG,YG,NPI,XI,YI,NP,MP,D)
C
C     THIS SUBROUTINE DETERMINES THE LCCAL INTERPCLATION POINTS FOR THE
C     GRID VERSION OF FRANKE'S METHOD OF SURFACE INTERPOLATION.
C     MINPTS POINTS ARE REQUIRED FOR EACH REGION.
C     IF FEWER THAN MINPTS POINTS ARE FCUND IN THE REGION, THE NEXT
C     CLOSEST POINTS (IN THE SUP NORM AFTER THE CURRENT RECTANGLE IS
C     TRANSFORMED ONTO (0,1)) ARE USED.  MINPTS IS SET TO 3, WHICH IS
C     THE RECOMMENDED VALUE, ALTHOUGH IT MAY BE ALTERED.
C
C     THE ARGUMENTS ARE AS FOLLOWS.
C
C        NXG   - INPUT.  NUMBER OF VERTICAL GRID LINES.
C        XG    - INPUT.  THE COORDINATES OF THE VERTICAL GRID LINES.
C        NYG   - INPUT.  NUMBER OF HORIZCNTAL GRID LINES.
C        YG    - INPUT.  THE COORDINATES OF THE HORIZONTAL GRID LINES.
C        NPI   - INPUT.  THE NUMBER OF DATA POINTS.
C        XI    -
C        YI    - INPUT.  THE DATA POINTS (XI,YI,FI), I=1,....,NPI.
C        FI    -
C        NP    - OUTPUT.  AN ARRAY WHICH GIVES THE INITIAL SUBSCRIPT IN
C                         THE ARRAY MP AT WHICH THE SUBSCRIPTS FOR THE
C                         LOCAL INTERPOLATION POINTS ARE STORED.
C        MP    - OUTPUT.  AN ARRAY WHICH GIVES THE SUBSCRIPTS FOR THE
C                         LOCAL INTERPOLATION POINTS.
C        D     - A WORK ARRAY OF DIMENSICN AT LEAST NPI.
C
      DIMENSION XG(1), YG(1), XI(1), YI(1), NP(1), MP(1), D(1)
      DATA MINPTS/3/
      IJ = 1
      NP(1) = 1
      L = 0
C
      DO 200 J=1,NYG
      YGA = (YG(J+2)+YG(J))/2.
      DYG = YG(J+2)-YG(J)
C
      DO 180 I=1,NXG
      XGA = (XG(I+2)+XG(I))/2.
      DXG = XG(I+2)-XG(I)
      IJ = IJ+1
C
C     DETERMINE THE POINTS IN THE (I,J)TH RECTANGLE.
C
C
      DC 120 NK=1,NPI
      IF (XI(NK).GT.XG(I+2).OR.XI(NK).LT.XG(I)) GC TO 100
      IF (YI(NK).GT.YG(J+2).OR.YI(NK).LT.YG(J)) GC TO 100
```

```
      L = L+1
      D(NK) = 1.E10
      MP(L) = NK
      GC TO 120
  100 D(NK) = AMAX1(ABS(XI(NK)-XGA)/DXG,ABS(YI(NK)-YGA)/DYG)
  120 CCNTINUE
C
      NP(IJ) = L+1
      IF (NP(IJ)-NP(IJ-1).GE.MINPTS) GO TO 180
C
C     ADD THE CLOSEST PCINTS IF THERE ARE LESS THAN MINPTS IN THE
C     RECTANGLE.
C
      LM = MINPTS-(NP(IJ)-NP(IJ-1))
C
      DC 160 II=1,LM
      L = L+1
      MP(L) = 1
      DM = D(1)
C
      DO 140 NK=2,NPI
      IF (D(NK).GE.DM) GC TO 140
      DM = D(NK)
      MP(L) = NK
  140 CONTINUE
C
      NK = MP(L)
  160 D(NK) = 1.E10
C
      NP(IJ) = L+1
  180 CCNTINUE
C
  200 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE CLOB22 (MODL,XI,YI,FI,NXG,XG,NYG,YG,NP,MP,AL,AB,IER)
C
C     THIS SUBROUTINE CCNSTRUCTS THE LOCAL APPROXIMANTS FOR THE GRID
C     VERSION OF FRANKE'S METHOD.  THE LCCAL APPRCXIMATIONS MAY BE
C     EITHER OPTIMAL APPROXIMATIONS IN B CORNER 2,2, OR OPTIMAL
C     APPROXIMATIONS IN B CCRNER 2,2 BOCLEAN SUM THE LEAST SQUARES
C     PLANE, OR FOR APPROXIMATION, RATHER THAN INTERPOLATION
C     THE LEAST SQUARES PLANE MAY BE SPECIFIED.
C
C     THE ARGUMENTS ARE AS FOLLOWS.
C
C        MODL   - INPUT.  SPECIFIES THE TYPE OF LOCAL APPROXIMATION
C                         DESIRED.
C                       = 1,  USE THE OPTIMAL APPRCXIMATION IN B CORNER
C                             2,2.
C                       = 2,  USE THE LEAST SQUARES PLANE.
C                       = 3,  USE THE OPTIMAL APPRCXIMATION IN B CORNER
C                             2,2 BOOLEAN SUM THE LEAST SQUARES PLANE.
C        XI     -
C        YI     - INPUT.  THE DATA POINTS (XI,YI,FI),I=1,NPI.
C        FI     -
C        NXG    - INPUT.  THE NUMBER OF VERTICAL GRID LINES.
C        XG     - INPUT.  THE COORDINATES OF THE VERTICAL GRID LINES.
C        NYG    - INPUT.  THE NUMBER OF HORIZONTAL GRID LINES.
C        YG     - INPUT.  THE COORDINATES OF THE HORIZONTAL GRID LINES.
C        NP     - INPUT.  AN ARRAY WHICH GIVES THE INITIAL SUBSCRIPT IN
C                         THE ARRAY MP AT WHICH THE SUBSCRIPTS FOR THE
C                         LOCAL INTERPOLATION POINTS ARE STORED.
C        MP     - INPUT.  AN ARRAY WHICH GIVES THE SUBSCRIPTS FOR THE
C                         LOCAL INTERPOLATION POINTS.
C        AL     - INPUT.  THE COEFFICIENTS FOR THE LINEAR LEAST SQUARES
C                         FIT, WHEN MODL = 2 OR 3.
C        AB     - INPUT.  THE COEFFICIENTS FOR THE CPTIMAL APPROXIMATION
C                         WHEN MODL = 1 OR 3.
C        IER    - OUTPUT.  RETURN INDICATOR.
C                   = 0,  NORMAL RETURN.
C                   = 1,  SINGULAR MATRIX HAS BEEN DETECTED IN THE
C                         LEAST SQUARES FIT.
C                   = 2,  SINGULAR MATRIX HAS BEEN DETECTED IN THE
C                         OPTIMAL FIT.
C                             IN CASE OF A SINGULAR MATRIX, THE GRID
C                             VALUE (I,J) AND THE CATA POINTS ASSOCIATED
C                             WITH THAT RECTANGLE ARE PRINTED.
C                   = 3,  THE NUMBER OF POINTS ASSOCIATED WITH SOME
C                         RECTANGLE I,J IS BIGGER THAN PRESENTLY PERMIT-
C                         TED.  THE ARRAY C MUST BE DIMENSIONED
C                         NC*(NC+3)/2.
C
```

```
C
C     BECAUSE OF THE SHORT WORD LENGTH OF THE IBM 360/370 COMPUTERS
C     CERTAIN VARIABLES ARE DECLARED AS DOUBLE PRECISION.  THIS STATE-
C     MENT MAY BE SAFELY REPLACED WITH THE STATEMENT  ' REAL K ' WHEN
C     THIS PROGRAM IS USED ON COMPUTERS WITH LONGER WORD LENGTHS.
C
      DOUBLE PRECISION K,D1,D2,C
      DIMENSION XI(1), YI(1), FI(1), NP(1), MP(1), AL(1), AB(1), XG(1),
     1YG(1), C(230)
      DATA NOUT,NC/6,20/
      IER = 0
      IJ = 0
      B = 1.
C
      DO 260 J=1,NYG
      DY = YG(J+2)-YG(J)
      A = 1.
C
      DO 240 I=1,NXG
      DX = XG(I+2)-XG(I)
      IJ = IJ+1
      LEND = NP(IJ+1)-NP(IJ)
      IF (LEND.GT.NC) GO TO 360
      LBS = LEND*(LEND+1)/2
      IALS = (IJ-1)*3
      IF (MODL.EQ.1) GO TO 160
C
C     CALCULATE THE LEAST SQUARES PLANE
C
C
      DO 100 LL=1,9
  100 C(LL) = 0.
C
      C(1) = LEND
C
      DO 120 L=1,LEND
      MPSL = NP(IJ)+L-1
      KL = MP(MPSL)
      XKL = XI(KL)-XG(I+1)
      YKL = YI(KL)-YG(J+1)
      C(2) = C(2)+XKL
      C(3) = C(3)+XKL**2
      C(4) = C(4)+YKL
      C(5) = C(5)+YKL*XKL
      C(6) = C(6)+YKL**2
      C(7) = C(7)+FI(KL)
      C(8) = C(8)+FI(KL)*XKL
  120 C(9) = C(9)+FI(KL)*YKL
```

```
C
      CALL LEQT1P (C,1,3,C(7),1,0,D1,D2,KER)
      IF (KER.NE.0) GO TO 280
C
      DO 140 LL=1,3
      IAL = IALS+LL
  140 AL(IAL) = C(LL+6)
C
  160 IF (MODL.EQ.2) GO TO 240
C
C     CALCULATE THE B CORNER 2,2 OPTIMAL APPROXIMATION
C
      KK = 0
C
      DO 200 LI=1,LEND
      MPI = NP(IJ)+LI-1
      KI = MP(MPI)
      XKI = (XI(KI)-XG(I))/DX
      YKI = (YI(KI)-YG(J))/DY
C
      DO 180 LJ=1,LI
      MPJ = NP(IJ)+LJ-1
      KJ = MP(MPJ)
      XKJ = (XI(KJ)-XG(I))/DX
      YKJ = (YI(KJ)-YG(J))/DY
      KK = KK+1
  180 C(KK) = K(A,B,XKI,YKI,XKJ,YKJ)
C
      LB = LBS+LI
      C(LB) = FI(KI)
      IF (MODL.EQ.1) GO TO 200
      XIK = XI(KI)-XG(I+1)
      YIK = YI(KI)-YG(J+1)
      C(LB) = C(LB)-(AL(IALS+1)+AL(IALS+2)*XIK+AL(IALS+3)*YIK)
  200 CONTINUE
C
      CALL LEQT1P (C,1,LEND,C(LBS+1),1,0,D1,D2,KER)
      IF (KER.NE.0) GO TO 300
C
      DO 220 LI=1,LEND
      IAB = NP(IJ)+LI-1
      LB = LBS+LI
  220 AB(IAB) = C(LB)
C
      A = 0.
  240 CONTINUE
C
  260 B = 0.
```

```
C
      RETURN
C
C     ERROR RETURNS
C
  280 IER = 1
      GC TO 320
  300 IER = 2
  320 WRITE (NOUT,1) I,J,IER
      MPST = NP(IJ)
      MPSE = MPST+LEND-1
C
      DO 340 MPS=MPST,MPSE
      I = MP(MPS)
  340 WRITE (NOUT,2) I,XI(I),YI(I),FI(I)
C
      RETURN
  360 IER = 3
      WRITE (NOUT,3) I,J,LEND,NC
      RETURN
C
    1 FORMAT (46H0SINGULAR MATRIX DETECTED FOR GRID POINT I,J =,2I5,12H
     1,     IER = ,I2//40H THE DATA POINTS INVOLVEC ARE AS FOLLOWS/)
    2 FORMAT (1X,I5,3E15.6)
    3 FORMAT (15H0THE RECTANGLE ,I3,1H,,I3,5H HAS ,I3,27H POINTS ASSOCIA
     1TED WITH IT /51H THE CURRENT LIMIT IMPOSED IN SUBRCUTINE CLCB22 IS
     2 ,I3/84H THIS MAY BE MODIFIED BY REDEFINING NC AND CHANGING THE DI
     3MENSION OF C APPROPRIATELY)
      END
```

```
      SUBROUTINE EVLB22 (MODL,XI,YI,NXG,XG,NYG,YG,NP,MP,AL,AB,
     1                   NXO,XO,NYO,YO,FO)
C
C     THIS SUBROUTINE EVALUATES THE INTERPOLANT FCR THE GRID VERSION OF
C     FRANKE'S METHOD.  THE FUNCTION IS EVALUATED AT THE GRID OF POINTS
C     INDICATED BY NXO, XO, NYO, YO, ANC THESE VALUES ARE RETURNED
C     IN THE ARRAY FO, WHICH IS ASSUMED TO BE DIMENSIONED (NXO,NYO).
C
C     THE ARGUMENTS ARE AS FOLLOWS.
C
C        MODL  - INPUT.   SPECIFIES THE TYPE OF LCCAL APPROXIMATION USED.
C                  = 1,   USED THE OPTIMAL APPROXIMATION IN B CORNER 2,2.
C                  = 2,   USED THE LEAST SQUARES PLANE.
C                  = 3,   USED THE OPTIMAL APPROXIMATION IN B CORNER 2,2
C                         BOOLEAN SUM THE LEAST SQUARES PLANE.
C        XI    -
C        YI    - INPUT.   THE DATA POINTS (XI,YI,FI),I=1,...,NPI.
C        FI    -
C        NXG   - INPUT.   THE NUMBER OF VERTICAL GRID LINES.
C        XG    - INPUT.   THE COORDINATES OF THE VERTICAL GRID LINES.
C        NYG   - INPUT.   THE NUMBER OF HORIZONTAL GRID LINES.
C        YG    - INPUT.   THE COORDINATES OF THE HORIZONTAL GRID LINES.
C        NP    - INPUT.   AN ARRAY WHICH GIVES THE INITIAL SUBSCRIPT IN
C                         THE ARRAY MP AT WHICH THE SUBSCRIPTS FOR THE
C                         LOCAL INTERPOLATION POINTS ARE STORED.
C        MP    - INPUT.   AN ARRAY WHICH GIVES THE SUBSCRIPTS FOR THE
C                         LOCAL INTERPOLATION POINTS.
C        AL    - INPUT.   THE COEFFICIENTS FOR THE LEAST SQUARES PLANE,
C                         WHEN MODL = 2 OR 3.
C        AB    - INPUT.   THE COEFFICIENTS FOR THE CPTIMAL APPROXIMATION
C                         WHEN MODL = 1 OR 3.
C        NXO   - INPUT.   THE NUMBER OF XC VALUES AT WHICH THE INTERPO-
C                         LATION FUNCTION IS TO BE CALCULATED.
C        XO    - INPUT.   THE VALUES OF X AT WHICH THE INTERPOLATION
C                         FUNCTION IS TC BE CALCULATED.
C        NYO   - INPUT.   THE NUMBER OF YO VALUES AT WHICH THE INTERPO-
C                         LATION FUNCTION IS TO BE CALCULATED.
C        YO    - INPUT.   THE VALUES OF Y AT WHICH THE INTERPOLATION
C                         FUNCTION IS TO BE CALCULATED.
C        FO    - OUTPUT.   VALUES OF THE INTERPOLATION FUNCTION AT THE
C                         GRID POINTS INDICATED BY NXC, XO, NYO, YO.
C                         FO IS ASSUMED TO BE DIMENSICNED (NXO,NYO) IN THE
C                         CALLING PROGRAM.
C
      DIMENSION XG(1), YG(1), XI(1), YI(1), NP(1), MP(1), FC(4), AL(1),
     1AB(1), XO(1), YO(1), FO(NXO,1)
C
C     BECAUSE OF THE SHORT WORD LENGTH OF THE IBM 360/370 COMPUTERS
```

```
C
C     CERTAIN VARIABLE ARE DECLARED AS DOUBLE PRECISION.  THIS STATE-
C     MENT MAY BE SAFELY REPLACED WITH THE STATEMENT  ' REAL K ' WHEN
C     THIS PROGRAM IS USED ON COMPUTERS WITH LONGER WORD LENGTHS.
C
      DOUBLE PRECISION K
C
C     ARITHMETIC STATEMENT FUNCTION FOR THE HERMITE QUINTIC.
C
      H5(S) = 1.-S**3*((6.*S-15.)*S+10.)
      J = 1
C
      DO 640 JO=1,NYO
C
C     DETERMINE THE LOCATION OF THE POINT YO IN TERMS OF THE SMALLEST
C     VALUE OF J SUCH THAT YO(JO) IS IN SOME RECTANGLE (I,J).
C
      YV = YO(JO)
      JJS = J+1
      IF (YV.LT.YG(JJS)) JJS=1
C
      DO 100 JJ=JJS,NYG
      IF (YV.LT.YG(JJ+1)) GO TO 120
  100 CONTINUE
C
      J = NYG
      GO TO 140
  120 J = JJ-1
  140 JD = 3
      IF (J.GE.1) GO TO 160
      JD = 0
      J = 1
      GO TO 180
  160 IF (J.LT.NYG) GO TO 180
      JD = 6
  180 DY = YG(J+2)-YG(J+1)
      I = 1
C
      DO 620 IO=1,NXO
C
C     DETERMINE THE LOCATION OF THE POINT XO IN TERMS OF THE SMALLEST
C     VALUE OF I SUCH THAT XO(IO) IS IN THE RECTANGLE (I,J).
C
      IIS = I+1
      XV = XO(IO)
      IF (XV.LT.XG(IIS)) IIS=1
C
      DO 200 II=IIS,NXG
      IF (XV.LT.XG(II+1)) GO TO 220
```

```
  200 CONTINUE
C
      I = NXG
      GO TO 240
  220 I = II-1
  240 ID = 2
      IF (I.GE.1) GO TO 260
      IC = 1
      I = 1
      GO TO 280
  260 IF (I.LT.NXG) GO TO 280
      ID = 3
  280 DX = XG(I+2)-XG(I+1)
      KD = ID+JD
      A = 0.
      IF (I.EQ.1) A = 1.
      B = 0.
      IF (J.EQ.1) B = 1.
      GO TO (300,360,300,440,520,440,300,360,300), KD
C
C     THIS IS FOR (XO(IO),YO(JO)) POINTS IN A SINGLE RECTANGLE (I,J)
C
  300 FV = 0.
      IJ = (J-1)*NXG+I
      IAL = 3*IJ-2
      IF (MODL.EQ.2) GO TO 340
      LMAX = NP(IJ+1)-NP(IJ)
      DXA = XG(I+2)-XG(I)
      DYA = YG(J+2)-YG(J)
      XVD = (XV-XG(I))/DXA
      YVD = (YV-YG(J))/DYA
C
      DO 320 L=1,LMAX
      MPS = NP(IJ)+L-1
      KI = MP(MPS)
      XKI = (XI(KI)-XG(I))/DXA
      YKI = (YI(KI)-YG(J))/DYA
  320 FV = FV+AB(MPS)*K(A,B,XKI,YKI,XVD,YVD)
C
  340 IF (MODL.NE.1) FV = FV+AL(IAL)+AL(IAL+1)*(XV-XG(I+1))+AL(IAL+2)*(Y
     1V-YG(J+1))
      GO TO 620
C
C     THIS IS FOR XO(IO),YO(JO)) POINTS WHICH ARE IN TWO RECTANGLES,
C     (I,J) AND (I+1,J).
C
  360 DYA = YG(J+2)-YG(J)
      YVD = (YV-YG(J))/DYA
```

```
C
      DO 420 IP=1,2
      FC(IP) = 0.
      IS = I+IP-1
      IJ = (J-1)*NXG+IS
      IAL = 3*IJ-2
      IF (MODL.EQ.2) GO TO 400
      DXA = XG(IS+2)-XG(IS)
      XVD = (XV-XG(IS))/DXA
      LMAX = NP(IJ+1)-NP(IJ)
C
      DO 380 L=1,LMAX
      MPS = NP(IJ)+L-1
      KI = MP(MPS)
      XKI = (XI(KI)-XG(IS))/DXA
      YKI = (YI(KI)-YG(J))/DYA
  380 FC(IP) = FC(IP)+AB(MPS)*K(A,B,XKI,YKI,XVD,YVD)
C
  400 IF (MODL.NE.1) FC(IP)=FC(IP)+AL(IAL)+AL(IAL+1)*(XV-XG(IS+1))+AL(IA
     1L+2)*(YV-YG(J+1))
      A = 0.
  420 CONTINUE
C
      WI = H5((XV-XG(I+1))/DX)
      FV = FC(1)*WI+(1.-WI)*FC(2)
      GO TO 620
C
C     THIS IS FOR (XO(IO),YO(JO)) POINTS WHICH ARE IN TWO RECTANGLES,
C     (I,J) AND (I,J+1).
C
  440 DXA = XG(I+2)-XG(I)
      XVD = (XV-XG(I))/DXA
C
      DO 500 JP=1,2
      FC(JP) = 0.
      JS = J+JP-1
      IJ = (JS-1)*NXG+I
      IAL = 3*IJ-2
      IF (MODL.EQ.2) GO TO 480
      DYA = YG(JS+2)-YG(JS)
      YVD = (YV-YG(JS))/DYA
      LMAX = NP(IJ+1)-NP(IJ)
C
      DO 460 L=1,LMAX
      MPS = NP(IJ)+L-1
      KJ = MP(MPS)
      XKJ = (XI(KJ)-XG(I))/DXA
      YKJ = (YI(KJ)-YG(JS))/DYA
```

```
  460 FC(JP) = FC(JP)+AB(MPS)*K(A,B,XKJ,YKJ,XVD,YVD)
C
  480 IF (MODL.NE.1) FC(JP)=FC(JP)+AL(IAL)+AL(IAL+1)*(XV-XG(I+1))+AL(IAL
     1+2)*(YV-YG(JS+1))
      B = 0.
  500 CONTINUE
C
      UJ = H5((YV-YG(J+1))/DY)
      FV = FC(1)*UJ+(1.-UJ)*FC(2)
      GO TO 620
C
C     THIS IS FOR (XO(IO),YO(JO)) POINTS WHICH ARE IN FOUR RECTANGLES,
C     (I,J), (I+1,J), (I,J+1), AND (I+1,J+1).
C
  520 KFC = 0
C
      DO 600 JP=1,2
      JS = J+JP-1
      DYA = YG(JS+2)-YG(JS)
      YVD = (YV-YG(JS))/DYA
      A = 1.
C
      DO 580 IP=1,2
      IS = I+IP-1
      IJ = (JS-1)*NXG+IS
      IAL = 3*IJ-2
      KFC = KFC+1
      FC(KFC) = 0.
      IF (MODL.EQ.2) GO TO 560
      IF (IS.GT.1) A = 0.
      DXA = XG(IS+2)-XG(IS)
      XVD = (XV-XG(IS))/DXA
      LMAX = NP(IJ+1)-NP(IJ)
C
      DO 540 L=1,LMAX
      MPS = NP(IJ)+L-1
      KI = MP(MPS)
      XKI = (XI(KI)-XG(IS))/DXA
      YKI = (YI(KI)-YG(JS))/DYA
  540 FC(KFC) = FC(KFC)+AB(MPS)*K(A,B,XKI,YKI,XVD,YVD)
C
  560 IF (MODL.NE.1) FC(KFC)=FC(KFC)+AL(IAL)+AL(IAL+1)*(XV-XG(IS+1))+AL(
     1IAL+2)*(YV-YG(JS+1))
  580 CONTINUE
C
      B = 0.
  600 CONTINUE
C
```

```
      WI = H5((XV-XG(I+1))/DX)
      UJ = H5((YV-YG(J+1))/DY)
      FV = WI*(UJ*FC(1)+(1.-UJ)*FC(3))+(1.-WI)*(UJ*FC(2)+(1.-UJ)*FC(4))
  620 FO(IO,JO) = FV
C
  640 CONTINUE
C
      RETURN
      END
```

```
      FUNCTION K (A,B,U,V,S,T)
C
C     THIS FUNCTION EVALUATES THE REPRESENTER FOR THE POINT EVALUATION
C     (AT (U,V)) FUNCTIONAL FOR THE SARD CORNER SPACE B CORNER 2,2.
C
C     THE ARGUMENTS ARE AS FOLLOWS.
C
C        A,B - INPUT.  THE BASE POINT (A,B) FOR THE SARD SPACE B
C                      CORNER 2,2.
C        U,V - INPUT.  THE LINEAR FUNCTIONAL IS EVALUATION AT (U,V).
C        S,T - INPUT.  THE REPRESENTER IS EVALUATED AT (S,T).
C
C     BECAUSE OF THE SHORT WORD LENGTH OF THE IBM 360/370 COMPUTERS
C     CERTAIN VARIABLES ARE DECLARED AS DOUBLE PRECISION.  THIS STATE-
C     MENT MAY BE SAFELY REPLACED WITH THE STATEMENT  ' REAL K ' WHEN
C     THIS PROGRAM IS USED ON COMPUTERS WITH LONGER WORD LENGTHS.
C
      DOUBLE PRECISION K,SMA,UMA,USMA,X,TRMS,GP1,GP2
      SS = S
      UU = U
      AA = A
      KVAR = 1
  100 SMA = SS-AA
      UMA = UU-AA
      USMA = SMA*UMA
      TRMS = 0.
      IF (USMA.LE.0.) GO TO 160
      IF (UMA.GE.0.) GO TO 120
      UMA = -UMA
      SMA = -SMA
  120 IF (SMA.LE.UMA) GO TO 140
      X = SMA
      SMA = UMA
      UMA = X
  140 TRMS = SMA/2.*(USMA-SMA**2/3.)
  160 GP2 = 1.+USMA+TRMS
      GO TO (180,200), KVAR
  180 GP1 = GP2
      SS = T
      UU = V
      AA = B
      KVAR = 2
      GO TO 100
  200 K = GP1*GP2
      RETURN
      END
```

```
THE VALUES OF MODE, KER, NIWK, AND NWK ARE     1     0    79    77

          FUNCTION VALUES
            2.557540                 1.156260                 0.343608
            1.686603                 0.729544                 0.211618
            0.737488                 0.343513                 0.093705

          DEVIATIONS (THESE VALUES REPRESENT ROUNDOFF ERROR)
           -0.000015                 0.000042                 0.000052
           -0.000009                 0.000020                 0.000050
            0.000009                 0.000038                 0.000009

THE VALUES OF MODE, KER, NIWK, AND NWK ARE     2     0    79    37

          FUNCTION VALUES
            2.658678                 1.522738                 0.682158
            1.754473                 0.877377                 0.284612
            0.933759                 0.413208                 0.109657

          DEVIATIONS (THESE VALUES REPRESENT ROUNDOFF ERROR)
            0.000003                 0.000002                -0.000001
            0.000001                 0.000000                -0.000000
            0.000001                -0.000000                 0.0

THE VALUES OF MODE, KER, NIWK, AND NWK ARE     3     0    79   104

          FUNCTION VALUES
            2.572663                 1.158037                 0.347955
            1.695004                 0.724120                 0.213975
            0.736699                 0.341232                 0.093384

          DEVIATIONS (THESE VALUES REPRESENT ROUNDOFF ERROR)
           -0.000018                 0.000011                 0.000047
           -0.000015                 0.000003                 0.000045
           -0.000008                 0.000040                 0.000058

THE VALUES OF MODE, KER, NIWK, AND NWK ARE     3     0    58    76

          FUNCTION VALUES
            2.376339                 1.107161                 0.348125
            1.626697                 0.747433                 0.217716
            0.711982                 0.324831                 0.097776

          DEVIATIONS (THESE VALUES REPRESENT ROUNDOFF ERROR)
            0.000092                -0.000019                -0.000096
            0.000033                -0.000010                -0.000066
            0.000026                 0.000023                 0.000017
```

REFERENCES

1. Hiroshi Akima, "A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Points", to appear in Transactions on Math Software.

2. R. E. Barnhill, "Representation and Approximation of Surfaces", in Mathematical Software III, pp. 69-120, John R. Rice, ed., Academic Press, 1977.

3. R. E. Barnhill and J. A. Gregory, "Polynomial Interpolation to Boundary Data on Triangles", Math Comp 29 (1975) 726-735.

4. R. Franke, "Locally Determined Smooth Interpolation at Irregularly Spaced Points in Several Variables", JIMA 19 (1977) 471-482.

5. R. Franke, "On the Computation of Optimal Approximations in Sard Corner Spaces $B_{\lceil p,q \rceil}$", Technical Report # NPS- 53Fe76121, Naval Postgraduate School, Monterey, CA (to appear in SIAM J. Numer Anal.).

6. R. Franke, "Reproducing Kernel Functions in Sard Corner Spaces", Technical Report # NPR-53-78-001, Naval Postgraduate School, Monterey, CA.

7. C. L. Lawson, "Software for C' Surface Interpolation", in Mathematical Software III, pp. 161-193, John R. Rice, ed., Academic Press, 1977.

8. L. L. Schumaker, "Fitting Surfaces to Scattered Data", in Approximation Theory II, pp. 203-268, edited by G. G. Lorentz, C. K. Chin, and L. L. Schumaker, Academic Press, 1976.

DISTRIBUTION LIST

Defense Documentation Center Cameron Station Alexandria, VA 22314	2
Dudley Knox Library Naval Postgraduate School Monterey, CA 93940	2
Dean of Research Naval Postgraduate School Monterey, CA 93940	2
Department of Mathematics	
C. O. Wilde, Chairman	1
Professor C. Comstock	1
Professor F. Faulkner	1
Professor A. Schoenstadt	1
Professor R. Franke	5
Naval Postgraduate School Monterey, CA 93940	
Dr. Richard Lau Office of Naval Research 1030 East Green St. Pasadena, CA 91106	1
Professor R. E. Barnhill Department of Mathematics University of Utah Salt Lake City, UT 84112	1
Professor G. M. Nielson Department of Mathematics Arizona State University Tempe, AZ 85281	1
Chief of Naval Research ATT: Mathematics Program Arlington, VA 22217	2
Rosemary E. Chang Sandia Laboratories Applied Mathematics, Division 2-8235 Livermore, CA 94550	1
Hiroshi Akima Office of Telecommunications Department of Commerce Boulder, CO 80302	1